Provable Tempered Overfitting of Minimal Nets and Typical Nets

Itamar Harel ITAMARHAREL01@GMAIL.COM
Technion

William M. Hoza
The University of Chicago

Gal Vardi
Weizmann Institute of Science

Itay Evron
Technion

Nathan Srebro
Toyota Technological Institute at Chicago

Daniel Soudry
Technion

Abstract

We study the overfitting behavior of fully connected deep Neural Networks (NNs) with binary weights fitted to perfectly classify a noisy training set. We consider interpolation using both the smallest NN (having the minimal number of weights) and a random interpolating NN. For both learning rules, we prove overfitting is tempered. Our analysis rests on a new bound on the size of a threshold circuit consistent with a partial function.

1. Introduction

Neural networks (NNs) famously exhibit strong generalization capabilities, seemingly in defiance of traditional generalization theory. Often, NNs generalize well even when trained to interpolate the training data *perfectly* [94]. This motivated an extensive line of work on overfitting and generalization in NNs trained to perfectly fit a dataset with corrupted labels (*e.g.,* [5, 28, 48, 57]).

To better understand generalization in NNs, Mallinar et al. [57] proposed a taxonomy of benign, tempered, and catastrophic overfitting. An algorithm that perfectly interpolates a training set with corrupted labels, *i.e.,* an interpolator, exhibits tempered overfitting if its generalization error is neither benign nor catastrophic — not optimal but much better than trivial. However, the characterization of overfitting in NNs is still incomplete, especially in *deep* NNs when the input dimension is neither very high nor very low. Our paper studies the overfitting behavior of deep NNs in this regime.

We analyze tempered overfitting in "min-size" NN interpolators, *i.e.,* whose layer widths are selected to minimize the total number of weights. A model's number of parameters is a natural complexity measure in learning theory and practice. For instance, it is theoretically well understood that L_1 regularization in a sparse linear regression setting yields a sparse regressor. Practically, finding small-sized deep models is a common objective used in pruning and neural architecture search (*e.g.,* [33, 54]). Recently, Manoj and Srebro [58] proved that the *shortest program* (Turing machine) that perfectly interpolates noisy datasets exhibits tempered overfitting, illustrating how a powerful model can avoid catastrophic overfitting by returning a min-size interpolator.

© I. Harel, W.M. Hoza, G. Vardi, I. Evron, N. Srebro & D. Soudry.

We further study tempered overfitting in random ("typical") interpolators — NNs sampled uniformly from the set of parameters that perfectly fit the training set. Given a narrow teacher model and no label noise, Buzaglo et al. [13] recently proved that typical interpolators, which may be *highly overparameterized*, generalize well, even without relying on explicit regularization or the implicit bias of any gradient algorithm.

For both min-size and typical NN interpolators, we study the generalization behavior under an underlying *noisy* teacher model. We focus on deep NNs with binary weights and activations (similar NNs are used in resource-constrained environments; *e.g.,* [42]). Our analysis reveals that these models exhibit a tempered overfitting behavior that depends on the label noise's statistical properties (*i.e.,* independent noise leads to milder bounds). Our results are illustrated in Figure 1 below.

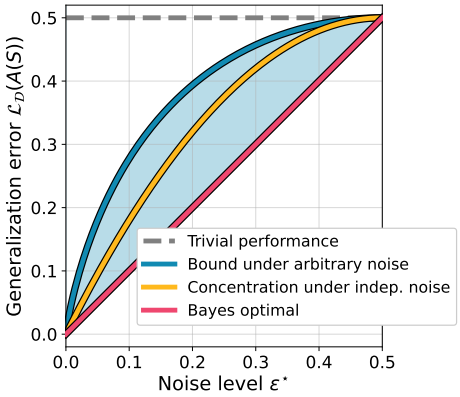


Figure 1: **Overfitting behaviors.** Consider a binary classification problem with an $\varepsilon^\star$-probability for label flips. Assuming balanced classes, the trivial generalization error is 0.5 (in gray; *e.g.,* with a constant predictor). A Bayes-optimal hypothesis (in red) obtains a generalization error of $\varepsilon^\star$. For large enough training sets, our results dictate tempered overfitting — for arbitrary noise, the error is approximately bounded by $1-2^{-H(\varepsilon^\star)}$, *i.e.,* $1-2^{\varepsilon^\star \log(\varepsilon^\star)+(1-\varepsilon^\star)\log(1-\varepsilon^\star)}$ (blue); and for independent noise, the error is concentrated around the tighter $2\varepsilon^\star(1-\varepsilon^\star)$ (orange). A similar figure was previously shown for shortest-program interpolators [58].

The contributions of this paper are:

- Returning a min-size NN interpolator is a natural learning rule that follows the Occam's-razor principle. We show that this learning rule exhibits tempered overfitting (Section 4.1).
- We prove that overparameterized random NN interpolators typically exhibit tempered overfitting with generalization close to a Bayes-optimal predictor (Section 4.2).
- To the best of our knowledge, ours are the first theoretical results on benign or tempered overfitting that: (1) apply to deep NNs, and (2) do not require a very high or very low input dimension.
- The above results rely on a key technical result — datasets generated by a constant-size teacher model with label noise can be interpolated[1] using a NN of constant depth with threshold activations, binary weights, a width sub-linear in N, and roughly $H(\varepsilon^\star)\cdot N$ weights, where $H(\varepsilon^\star)$ is the binary entropy function of the fraction of corrupted labels (Section 3).

2. Setting

Notation. Bold characters indicate vectors (lowercase) and matrices (uppercase). Regular uppercase characters indicate random variables (or vectors). We use $\oplus$ to denote the XOR of two binary values, and $\odot$ to denote the Hadamard (elementwise) product between two vectors. We use $\log = \log_2$ and $\mathrm{Ber}(p)$ to denote the Bernoulli distribution with probability p. Finally, $H(\mathcal{D})$ denotes the entropy of a distribution $\mathcal{D}$, and $H(\varepsilon) \triangleq H(\mathrm{Ber}(\varepsilon))$ denotes the binary entropy function.

1. As long as it has no repeated datapoints with opposite labels. See our Def. 3 of consistent datasets.

2.1. Model: Fully connected threshold NNs with binary weights

Similarly to Buzaglo et al. [13], we define the following model.

Definition 1 (Binary threshold networks) *For a depth L, widths $\underline{d} = (d_1, \ldots, d_L)$, input dimension d_0, a binary threshold network is a mapping $\boldsymbol{\theta} \mapsto h_{\boldsymbol{\theta}}$ such that $h_{\boldsymbol{\theta}} : \{0,1\}^{d_0} \to \{0,1\}$, parameterized by $\boldsymbol{\theta} = \left(\mathbf{W}^{(l)}, \mathbf{b}^{(l)}, \boldsymbol{\gamma}^{(l)}\right)_{l=1}^{L} \in \Theta^{BTN}(\underline{d})$, where for every layer $l \in [L]$,*

$$\mathbf{W}^{(l)} \in \mathcal{Q}_l^W = \{0,1\}^{d_l \times d_{l-1}}\,,\ \boldsymbol{\gamma}^{(l)} \in \mathcal{Q}_l^\gamma = \{-1,0,1\}^{d_l}\,,\ \mathbf{b}^{(l)} \in \mathcal{Q}_l^b = \{-d_{l-1}+1, \ldots, d_{l-1}\}^{d_l}\ .$$

This mapping is defined recursively as $h_{\boldsymbol{\theta}}(\mathbf{x}) = h^{(L)}(\mathbf{x})$ where $h^{(0)}(\mathbf{x}) = \mathbf{x}$ and

$$\forall l \in [L] \quad h^{(l)}(\mathbf{x}) = \mathbb{I}\left\{\left(\boldsymbol{\gamma}^{(l)} \odot \left(\mathbf{W}^{(l)} h^{(l-1)}(\mathbf{x})\right) + \mathbf{b}^{(l)}\right) > \mathbf{0}\right\}\ .$$

The total number of weights is $w(\underline{d}) = \sum_{l=1}^{L} d_l d_{l-1}$, and the number of neurons is $n(\underline{d}) = \sum_{l=1}^{L} d_l$. We denote the set of functions that can be implemented by binary threshold networks by $\mathcal{H}_{\underline{d}}^{\mathrm{BTN}}$.

2.2. Data model: A teacher network and label-flip noise

Data distribution. Let $\mathcal{X} = \{0,1\}^{d_0}$ and let $\mathcal{D}$ be some joint distribution over $\mathcal{X} \times \{0,1\}$.

Assumption 2 (Teacher assumption) *We assume a "teacher NN" $h^\star$ generating the labels. A label flipping noise is then added with a noise level of $\varepsilon^\star = \mathbb{P}_{(X,Y)\sim\mathcal{D}}(Y \neq h^\star(X))$. The label noise is* independent *when $Y \oplus h^\star(X)$ is independent of the features X.*

2.3. Learning problem: Classification with interpolators

We consider binary classification on a training set $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$ of N datapoints sampled from $\mathcal{D}$ (iid). For a hypothesis $h\colon \mathcal{X} \to \{0,1\}$, the generalization error is $\mathcal{L}_{\mathcal{D}}(h) \triangleq \mathbb{P}_{(X,Y)\sim\mathcal{D}}(h(X) \neq Y)$. The training error is $\mathcal{L}_S(h) \triangleq \frac{1}{N}\sum_{n=1}^{N} \mathbb{I}\{h(\mathbf{x}_n) \neq y_n\}$ and h is an *interpolator* when $\mathcal{L}_S(h) = 0$. In this paper, we are specifically interested in *consistent* datasets that can be perfectly fit.

Definition 3 (Consistent datasets) *A dataset $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$ is consistent if*

$$\forall i, j \in [N]\ \ \mathbf{x}_i = \mathbf{x}_j \Longrightarrow y_i = y_j\ .$$

In Section 4, we reveal a tempered overfitting behavior for (1) min-size NN interpolators and (2) sampled random NN interpolators, both falling under the following framework.

Framework 1 Learning interpolators

Input: A training set S.
Algorithm:
if S is consistent: **return** an interpolator $A(S) = h \in \mathcal{H}$ (such that $\mathcal{L}_S(h) = 0$)
else: **return** an arbitrary hypothesis $A(S) = h \in \mathcal{H}$ (*e.g.*, $h(\mathbf{x}) = 0, \forall \mathbf{x}$)

3. Interpolating a noisy training set

Our generalization results rely on a technical result that any consistent dataset from our noisy teacher model can be memorized by a "student" small enough to induce meaningful generalization bounds.

Theorem 4 (Memorizing a consistent dataset) *For any teacher $h^\star$ of depth $L^\star$ and dimensions $\underline{d}^\star$ and any consistent training set S generated from it, there exists an interpolating NN h (i.e., $\mathcal{L}_S(h) = 0$) of depth $L = \max\{L^\star, 14\} + 2$ and dimensions $\underline{d}$, such that the number of weights is*

$$w(\underline{d}) \leq w(\underline{d}^\star) + N \cdot H(\mathcal{L}_S(h^\star)) + 2n(\underline{d}^\star) N^{3/4} \text{polylog} N + O(d_0(d_0 + n(\underline{d}^\star)) \cdot \log N)$$

and the maximal width is $\underline{d}_{\max} \leq \underline{d}^\star_{\max} + N^{3/4} \cdot H(\mathcal{L}_S(h^\star)) \cdot \text{polylog}(N) + O(d_0 \cdot \log(N))$.

We outline a brief proof idea in Appendix A.1 and give the complete proof in Appendix E.

4. Tempered overfitting of min-size and random interpolators

We study overfitting in two types of interpolating NNs. In both cases, we prove tempered overfitting and show that the test performance is not much worse than the Bayes optimal error.

In our results, we use the marginal peak probability of the distribution, as defined next.[2]

Definition 5 (Peak marginal probability) $\mathcal{D}_{\max} \triangleq \max_{x \in \mathcal{X}} \mathbb{P}_{(X,Y)\sim\mathcal{D}}(X = x)$.

4.1. Min-size interpolators

Min-size NN interpolators are networks of a certain depth with the fewest weights that interpolate a given training set. In realizable settings, the generalization of interpolating models with a restricted number of parameters follows from standard VC-bounds [4, 74]. However, interpolating *noisy* data requires larger models (to memorize noise [85]), thus making generalization guarantees challenging.

Learning rule: Min-size NN interpolators. Given a consistent set S and a fixed depth L, a min-size NN interpolator, or min-#weights interpolator, is a binary threshold network h (see Def. 1) that achieves $\mathcal{L}_S(h) = 0$ using a minimal number of weights. Recall that $w(\underline{d}) = \sum_{l=1}^{L} d_l d_{l-1}$ and define the *minimal* number of weights required to implement a given hypothesis h,

$$w_L(h) \triangleq \min_{\underline{d} \in \mathbb{N}^L} w(\underline{d}) \text{ s.t. } h \in \mathcal{H}^{\text{BTN}}_{\underline{d}}.$$

The learning rule is then defined as $A_L(S) \in \text{argmin}_h w_L(h) \text{ s.t. } \mathcal{L}_S(h) = 0$.

Theorem 6 *Let $\mathcal{D}$ be a distribution induced by a noisy teacher of depth $L^\star$, widths $\underline{d}^\star$, $n(\underline{d}^\star)$ neurons, and a noise level of $\varepsilon^\star < 1/2$ (Assumption 2). Let $S \sim \mathcal{D}^N$ be a training set such that $N = \omega\big(n(\underline{d}^\star)^4 \text{polylog}(n(\underline{d}^\star)) + d_0^2 \log d_0\big)$ and $N = o(\sqrt{1/\mathcal{D}_{\max}})$. Then, for any fixed depth $L \geq \max\{L^\star, 14\} + 2$, the generalization error of the min-size depth-L NN interpolator satisfies:*

- ***Under arbitrary label noise,*** $\mathbb{E}_S[\mathcal{L}_{\mathcal{D}}(A_L(S))] \leq 1 - 2^{-H(\varepsilon^\star)} + o(1)$.
- ***Under independent label noise,*** $|\mathbb{E}_S[\mathcal{L}_{\mathcal{D}}(A_L(S))] - 2\varepsilon^\star(1-\varepsilon^\star)| = o(1)$.

Here, $o(1)$ indicates terms that become insignificant when the number of samples N is large. We illustrate these behaviors in Figure 1. Moreover, we discuss these results and the proof idea in Appendix A.2. The complete proof with detailed characterization of the $o(1)$ terms is given in Appendix G.1.

2. We focus on cases where $N = o\left(1/\sqrt{\mathcal{D}_{\max}}\right)$, in which our noisy datasets are most likely consistent.

4.2. Random NN interpolators (posterior sampling)

Recent empirical [20, 84] and theoretical [13] works found that randomly sampled deep NNs interpolating a training set often generalize surprisingly well. We analyze such random interpolators under our *noisy* teacher model and reveal once again a tempered overfitting behavior.

Learning rule: Posterior sampling. For any consistent training set S, we define a posterior distribution over hypotheses $\mathcal{P}_S = \text{Uniform}\left\{\boldsymbol{\theta} \in \Theta^{\text{BTN}}(\underline{d}) \mid \mathcal{L}_S(h_{\boldsymbol{\theta}}) = 0\right\}$, *i.e.,* the distribution induced by the uniform distribution over parameters of NN interpolators of S. For inconsistent S we use an arbitrary $\mathcal{P}_S$. Then, the posterior sampling rule is $A_{\underline{d}}(S) \sim \mathcal{P}_S$.

Theorem 7 *Let $\mathcal{D}$ be a distribution induced by a noisy teacher of depth $L^\star$, widths $\underline{d}^\star$, $n(\underline{d}^\star)$ neurons, and a noise level of $\varepsilon^\star < 1/2$ (Assumption 2). Let $S \sim \mathcal{D}^N$ be a training set such that $N = \omega\big(n(\underline{d}^\star)^4 \operatorname{polylog}(n(\underline{d}^\star)) + d_0^2 \log d_0\big)$ and $N = o(\sqrt{1/\mathcal{D}_{\max}})$. Then, there exist constants $c_1, c_2 > 0$ such that for any student NN of depth $L \geq \max\{L^\star, 14\} + 2$ and widths $\underline{d} \in \mathbb{N}^L$ holding*

$$\forall l = 1, \ldots, L^\star - 1 \quad d_l \geq d_l^\star + N^{3/4} \cdot (\log N)^{c_1} + c_2 \cdot d_0 \cdot \log(N) \ , \tag{1}$$

the generalization error of posterior sampling satisfies the following.

- ***Under arbitrary label noise,***
$$\mathbb{E}_{S, A_{\underline{d}}(S)}\left[\mathcal{L}_{\mathcal{D}}\left(A_{\underline{d}}(S)\right)\right] \leq 1 - 2^{-H(\varepsilon^\star)} + O\left(\frac{n(\underline{d}) \cdot \log(\underline{d}_{\max} + d_0)}{N}\right) .$$
- ***Under independent label noise,***
$$\left|\mathbb{E}_{S, A_{\underline{d}}(S)}\left[\mathcal{L}_{\mathcal{D}}\left(A_{\underline{d}}(S)\right)\right] - 2\varepsilon^\star(1-\varepsilon^\star)\right| \leq O\left(\sqrt{\frac{n(\underline{d}) \cdot \log(\underline{d}_{\max} + d_0)}{N}}\right) .$$

The proof and a detailed description of the error terms are given in Appendix G.2. Remarkably, unlike standard bounds [4, 74], the bounds are non-trivial even for NNs which may be overparmeterized.

5. Discussion

We discuss related work in detail in Appendix A.3. Specifically, we elaborate on benign and tempered overfitting, and discuss connections between our work and the literature on memorization, posterior sampling, and the guess-and-check algorithm [20]. Furthermore, we position our results from Section 3 within existing literature on circuit complexity.

In this work, we focused on binary (fully connected) threshold networks of depth $L \geq 16$ (Section 2.1) with binary input features (Section 2.2), for which we were able to derive nontrivial generalization bounds. Using simple modifications, our results can be extended to other settings like NNs with higher weight quantization (see Remark 9), or ReLU networks (since any threshold network with binary weights can be computed by a not-much-larger ReLU network with a constant quantization level). Unfortunately, without more sophisticated arguments these extensions result in looser generalization bounds. The "bottleneck" is our reliance on tight (enough) bounds on the widths of interpolating NNs. Extending the results to other architectures (*e.g.,* CNNs, fully connected without neuron scaling, shallower models, etc.) and other quantization schemes (*e.g.,* floating point representations) will require utilizing their structure to derive tighter bounds on the complexity (*e.g.,* number of weights) needed to interpolate consistent datasets.

Our paper focused on consistent training sets (Def. 3) allowing perfect interpolation. Realistically, models do not always perfectly interpolate the training set, and it is interesting to find generalization bounds for non-interpolating models, depending on the training error. In addition, it is interesting to relate the generalization error to the training *loss*, and not just to the training accuracy. Such extensions will require either broadening our generalization results or deriving new ones.

Acknowledgements

We thank Alexander Chashkin for generously providing English-language expositions of some results from his work [17] as well as some results from Lupanov's work [56] (personal communication). The research of DS was Funded by the European Union (ERC, A-B-C-Deep, 101039436). Views and opinions expressed are however those of the author only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency (ERCEA). Neither the European Union nor the granting authority can be held responsible for them. DS also acknowledges the support of the Schmidt Career Advancement Chair in AI. GV is supported by a research grant from the Center for New Scientists at the Weizmann Institute of Science. Part of this work was done as part of the NSF-Simons funded Collaboration on the Mathematics of Deep Learning. NS was partially supported by the NSF TRIPOD Institute on Data Economics Algorithms and Learning (IDEAL) and an NSF-IIS award.

References

[1] Noga Alon, László Babai, and Alon Itai. A fast and simple randomized parallel algorithm for the maximal independent set problem. *J. Algorithms*, 7(4):567–583, 1986. ISSN 0196-6774. doi: 10.1016/0196-6774(86)90019-2.

[2] Noga Alon, Oded Goldreich, Johan Hå stad, and René Peralta. Simple constructions of almost k-wise independent random variables. *Random Structures Algorithms*, 3(3):289–304, 1992. ISSN 1042-9832. doi: 10.1002/rsa.3240030308.

[3] Peter L Bartlett and Philip M Long. Failures of model-dependent generalization bounds for least-norm interpolation. *The Journal of Machine Learning Research*, 22(1):9297–9311, 2021.

[4] Peter L Bartlett, Nick Harvey, Christopher Liaw, and Abbas Mehrabian. Nearly-tight vc-dimension and pseudodimension bounds for piecewise linear neural networks. *Journal of Machine Learning Research*, 20(63):1–17, 2019.

[5] Peter L. Bartlett, Philip M. Long, Gábor Lugosi, and Alexander Tsigler. Benign overfitting in linear regression. *Proceedings of the National Academy of Sciences*, 117(48):30063–30070, 2020.

[6] Daniel Barzilai and Ohad Shamir. Generalization in kernel regression under realistic assumptions. In *Forty-first International Conference on Machine Learning*, 2024.

[7] Eric B Baum. On the capabilities of multilayer perceptrons. *Journal of complexity*, 4(3): 193–215, 1988.

[8] Daniel Beaglehole, Mikhail Belkin, and Parthe Pandit. On the inconsistency of kernel ridgeless regression in fixed dimensions. *SIAM Journal on Mathematics of Data Science*, 5(4):854–872, 2023. doi: 10.1137/22M1499819.

[9] Paul W. Beame, Stephen A. Cook, and H. James Hoover. Log depth circuits for division and related problems. *SIAM J. Comput.*, 15(4):994–1003, 1986. ISSN 0097-5397. doi: 10.1137/0215070.

[10] Mikhail Belkin, Daniel J Hsu, and Partha Mitra. Overfitting or perfect fitting? Risk bounds for classification and regression rules that interpolate. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[11] Mikhail Belkin, Daniel Hsu, and Ji Xu. Two models of double descent for weak features. *SIAM Journal on Mathematics of Data Science*, 2(4):1167–1180, 2020.

[12] Sébastien Bubeck, Ronen Eldan, Yin Tat Lee, and Dan Mikulincer. Network size and size of the weights in memorization with two-layers neural networks. In *Neural Information Processing Systems*, 2020.

[13] Gon Buzaglo, Itamar Harel, Mor Shpigel Nacson, Alon Brutzkus, Nathan Srebro, and Daniel Soudry. How uniform random weights induce non-uniform bias: Typical interpolating neural networks generalize with narrow teachers. In *International Conference on Machine Learning (ICML)*, 2024.

[14] Yuan Cao, Quanquan Gu, and Mikhail Belkin. Risk bounds for over-parameterized maximum margin classification on sub-gaussian mixtures. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[15] Yuan Cao, Zixiang Chen, Misha Belkin, and Quanquan Gu. Benign overfitting in two-layer convolutional neural networks. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[16] Ashok K. Chandra, Larry Stockmeyer, and Uzi Vishkin. Constant depth reducibility. *SIAM J. Comput.*, 13(2):423–439, 1984. ISSN 0097-5397. doi: 10.1137/0213028.

[17] Alexander Chashkin. On the realization of partial boolean functions. In *Proceedings of the 7th International Conference on Discrete Models in the Theory of Control Systems*, pages 390–404, 2006. In Russian.

[18] Niladri S. Chatterji and Philip M. Long. Finite-sample analysis of interpolating linear classifiers in the overparameterized regime. *Journal of Machine Learning Research*, 22(129):1–30, 2021.

[19] Niladri S Chatterji, Philip M Long, and Peter L Bartlett. The interplay between implicit bias and benign overfitting in two-layer linear networks. *Journal of machine learning research*, 23 (263):1–48, 2022.

[20] Ping-yeh Chiang, Renkun Ni, David Yu Miller, Arpit Bansal, Jonas Geiping, Micah Goldblum, and Tom Goldstein. Loss landscapes are all you need: Neural network generalization can be explained without the implicit bias of gradient descent. In *The Eleventh International Conference on Learning Representations*, 2023.

[21] Geoffrey Chinot and Matthieu Lerasle. On the robustness of the minimum ℓ_2 interpolator. *arXiv preprint arXiv:2003.05838*, 2020.

[22] Benny Chor and Oded Goldreich. On the power of two-point based sampling. *J. Complexity*, 5 (1):96–106, 1989. ISSN 0885-064X. doi: 10.1016/0885-064X(89)90015-0.

[23] Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27, 1967.

[24] Martin Dietzfelbinger. Universal hashing and k-wise independent random variables via integer arithmetic without primes. In *STACS 96 (Grenoble, 1996)*, volume 1046 of *Lecture Notes in Comput. Sci.*, pages 569–580. Springer, Berlin, 1996.

[25] Konstantin Donhauser, Nicolo Ruggeri, Stefan Stojanovic, and Fanny Yang. Fast rates for noisy interpolation require rethinking the effect of inductive bias. In *International Conference on Machine Learning (ICML)*, 2022.

[26] Wayne Eberly. Very fast parallel polynomial arithmetic. *SIAM J. Comput.*, 18(5):955–976, 1989. ISSN 0097-5397. doi: 10.1137/0218066.

[27] Guy Even, Oded Goldreich, Michael Luby, Noam Nisan, and Boban Veličković. Efficient approximation of product distributions. *Random Structures & Algorithms*, 13(1):1–16, 1998.

[28] Spencer Frei, Niladri S. Chatterji, and Peter L. Bartlett. Benign overfitting without linearity: Neural network classifiers trained by gradient descent for noisy linear data. In *Conference on Learning Theory (COLT)*, 2022.

[29] Spencer Frei, Gal Vardi, Peter Bartlett, and Nathan Srebro. Benign overfitting in linear classifiers and leaky relu networks from kkt conditions for margin maximization. In *Proceedings of Thirty Sixth Conference on Learning Theory*, volume 195 of *Proceedings of Machine Learning Research*, pages 3173–3228. PMLR, 12–15 Jul 2023.

[30] Erin George, Michael Murray, William Swartworth, and Deanna Needell. Training shallow relu networks on noisy data using hinge loss: when do we overfit and is it benign? *Advances in Neural Information Processing Systems*, 36, 2024.

[31] Nikhil Ghosh and Mikhail Belkin. A universal trade-off between the model size, test loss, and training loss of linear predictors. *SIAM Journal on Mathematics of Data Science*, 5(4): 977–1004, 2023.

[32] András Hajnal, Wolfgang Maass, Pavel Pudlák, Mário Szegedy, and György Turán. Threshold circuits of bounded depth. *J. Comput. System Sci.*, 46(2):129–154, 1993. ISSN 0022-0000. doi: 10.1016/0022-0000(93)90001-D.

[33] Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. *Advances in neural information processing systems*, 28, 2015.

[34] Trevor Hastie, Andrea Montanari, Saharon Rosset, and Ryan J Tibshirani. Surprises in high-dimensional ridgeless least squares interpolation. *Annals of statistics*, 50(2):949, 2022.

[35] Pooya Hatami and William Hoza. Paradigms for unconditional pseudorandom generators. *Foundations and Trends® in Theoretical Computer Science*, 16(1-2):1–210, 2024. ISSN 1551-305X. doi: 10.1561/0400000109.

[36] Alexander Healy and Emanuele Viola. Constant-depth circuits for arithmetic in finite fields of characteristic two. In *STACS 2006*, volume 3884 of *Lecture Notes in Comput. Sci.*, pages 672–683. Springer, Berlin, 2006. doi: 10.1007/11672142_55.

[37] Thomas Hofmeister, Walter Hohberg, and Susanne Köhling. Some notes on threshold circuits, and multiplication in depth 4. *Inform. Process. Lett.*, 39(4):219–225, 1991. ISSN 0020-0190. doi: 10.1016/0020-0190(91)90183-I.

[38] William M. Hoza and David Zuckerman. Simple optimal hitting sets for small-success **RL**. *SIAM J. Comput.*, 49(4):811–820, 2020. ISSN 0097-5397. doi: 10.1137/19M1268707.

[39] Guang-Bin Huang. Learning capability and storage capacity of two-hidden-layer feedforward networks. *IEEE transactions on neural networks*, 14(2):274–281, 2003.

[40] Guang-Bin Huang and Haroon A Babri. Upper bounds on the number of hidden neurons in feedforward networks with arbitrary bounded nonlinear activation functions. *IEEE transactions on neural networks*, 9(1):224–229, 1998.

[41] Shih-Chi Huang, Yih-Fang Huang, et al. Bounds on the number of hidden neurons in multilayer perceptrons. *IEEE transactions on neural networks*, 2(1):47–55, 1991.

[42] Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Binarized neural networks. *Advances in neural information processing systems*, 29, 2016.

[43] Nirmit Joshi, Gal Vardi, and Nathan Srebro. Noisy interpolation learning with shallow univariate reLU networks. In *The Twelfth International Conference on Learning Representations*, 2024.

[44] Peizhong Ju, Xiaojun Lin, and Jia Liu. Overfitting can be harmless for basis pursuit, but only to a degree. *Advances in Neural Information Processing Systems*, 33:7956–7967, 2020.

[45] Stasys Jukna. *Boolean function complexity*, volume 27 of *Algorithms and Combinatorics*. Springer, Heidelberg, 2012. ISBN 978-3-642-24507-7. doi: 10.1007/978-3-642-24508-4. Advances and frontiers.

[46] Kedar Karhadkar, Erin George, Michael Murray, Guido Montúfar, and Deanna Needell. Benign overfitting in leaky relu networks with moderate input dimension. *arXiv preprint arXiv:2403.06903*, 2024.

[47] Frederic Koehler, Lijia Zhou, Danica J. Sutherland, and Nathan Srebro. Uniform convergence of interpolators: Gaussian width, norm bounds and benign overfitting. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

[48] Guy Kornowski, Gilad Yehudai, and Ohad Shamir. From tempered to benign overfitting in relu neural networks. *Advances in Neural Information Processing Systems*, 36, 2024.

[49] Yiwen Kou, Zixiang Chen, Yuanzhou Chen, and Quanquan Gu. Benign overfitting for two-layer relu networks. *arXiv preprint arXiv:2303.04145*, 2023.

[50] Jianfa Lai, Manyun Xu, Rui Chen, and Qian Lin. Generalization ability of wide neural networks on $\mathbb{R}$. *arXiv preprint arXiv:2302.05933*, 2023.

[51] Tengyuan Liang and Alexander Rakhlin. Just interpolate: Kernel “ridgeless" regression can generalize. *Annals of Statistics*, 48(3):1329–1347, 2020.

[52] Tengyuan Liang and Benjamin Recht. Interpolating classifiers make few mistakes. *Journal of Machine Learning Research*, 24(20):1–27, 2023.

[53] Tengyuan Liang, Alexander Rakhlin, and Xiyu Zhai. On the multiple descent of minimum-norm interpolants and restricted lower isometry of kernels. In *Conference on Learning Theory (COLT)*, 2020.

[54] Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang. Learning efficient convolutional networks through network slimming. In *Proceedings of the IEEE international conference on computer vision*, pages 2736–2744, 2017.

[55] Oleg B. Lupanov. A method of circuit synthesis. *Izvesitya VUZ, Radiofizika*, 1, 1958. In Russian.

[56] Oleg B. Lupanov. On a certain approach to the synthesis of control systems – the principle of local coding. *Problemy Kibernetiki*, 14:31–110, 1965. In Russian.

[57] Neil Rohit Mallinar, James B Simon, Amirhesam Abedsoltan, Parthe Pandit, Misha Belkin, and Preetum Nakkiran. Benign, tempered, or catastrophic: Toward a refined taxonomy of overfitting. In *Advances in Neural Information Processing Systems*, 2022.

[58] Naren Sarayu Manoj and Nathan Srebro. Interpolation learning with minimum description length. *arXiv preprint arXiv:2302.07263*, 2023.

[59] Yishay Mansour, Noam Nisan, and Prasoon Tiwari. The computational complexity of universal hashing. *Theoret. Comput. Sci.*, 107(1):121–133, 1993. ISSN 0304-3975. doi: 10.1016/0304-3975(93)90257-T.

[60] Andrew D McRae, Santhosh Karnik, Mark Davenport, and Vidya K Muthukumar. Harmless interpolation in regression and classification with structured features. In *International Conference on Artificial Intelligence and Statistics*, pages 5853–5875. PMLR, 2022.

[61] Song Mei and Andrea Montanari. The generalization error of random features regression: Precise asymptotics and the double descent curve. *Communications on Pure and Applied Mathematics*, 2019.

[62] Xuran Meng, Difan Zou, and Yuan Cao. Benign overfitting in two-layer relu convolutional neural networks for xor data. *arXiv preprint arXiv:2310.01975*, 2023.

[63] Chris Mingard, Guillermo Valle-Pérez, Joar Skalse, and Ard A Louis. Is sgd a bayesian sampler? well, almost. *The Journal of Machine Learning Research*, 22(1):3579–3642, 2021.

[64] Andrea Montanari, Feng Ruan, Youngtak Sohn, and Jun Yan. The generalization error of max-margin linear classifiers: High-dimensional asymptotics in the overparametrized regime. *Preprint, arXiv:1911.01544*, 2020.

[65] Vidya Muthukumar, Kailas Vodrahalli, Vignesh Subramanian, and Anant Sahai. Harmless interpolation of noisy data in regression. *IEEE Journal on Selected Areas in Information Theory*, 2020.

[66] Vidya Muthukumar, Adhyyan Narang, Vignesh Subramanian, Mikhail Belkin, Daniel Hsu, and Anant Sahai. Classification vs regression in overparameterized regimes: Does the loss function matter? *Journal of Machine Learning Research*, 22(222):1–69, 2021.

[67] Jeffrey Negrea, Gintare Karolina Dziugaite, and Daniel Roy. In defense of uniform convergence: Generalization via derandomization with an application to interpolating predictors. In *International Conference on Machine Learning*, pages 7263–7272, 2020.

[68] Sejun Park, Jaeho Lee, Chulhee Yun, and Jinwoo Shin. Provable memorization via deep neural networks using sub-linear parameters. In *Conference on Learning Theory*, pages 3627–3661. PMLR, 2021.

[69] Nicholas Pippenger. The complexity of computations by networks. *IBM J. Res. Develop.*, 31 (2):235–243, 1987. ISSN 0018-8646. doi: 10.1147/rd.312.0235.

[70] Shashank Rajput, Kartik Sreenivasan, Dimitris Papailiopoulos, and amin karbasi. An exponential improvement on the memorization capacity of deep threshold networks. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

[71] Alexander Rakhlin and Xiyu Zhai. Consistency of interpolation with laplace kernels is a high-dimensional phenomenon. In *Conference on Learning Theory*, pages 2595–2623. PMLR, 2019.

[72] John H. Reif and Stephen R. Tate. On threshold circuits and polynomial computation. *SIAM J. Comput.*, 21(5):896–908, 1992. ISSN 0097-5397. doi: 10.1137/0221053.

[73] Michael A Sartori and Panos J Antsaklis. A simple method to derive bounds on the size and to train multilayer neural networks. *IEEE transactions on neural networks*, 2(4):467–471, 1991.

[74] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[75] Ohad Shamir. The implicit bias of benign overfitting. In *Conference on Learning Theory*, pages 448–478. PMLR, 2022.

[76] Claude E. Shannon. The synthesis of two-terminal switching circuits. *Bell System Tech. J.*, 28: 59–98, 1949. ISSN 0005-8580. doi: 10.1002/j.1538-7305.1949.tb03624.x.

[77] Kai-Yeung Siu and Jehoshua Bruck. On the power of threshold circuits with small weights. *SIAM J. Discrete Math.*, 4(3):423–435, 1991. ISSN 0895-4801. doi: 10.1137/0404038.

[78] Kai-Yeung Siu and Vwani P. Roychowdhury. On optimal depth threshold circuits for multiplication and related problems. *SIAM J. Discrete Math.*, 7(2):284–292, 1994. ISSN 0895-4801. doi: 10.1137/S0895480192228619.

[79] Kai-Yeung Siu, Jehoshua Bruck, Thomas Kailath, and Thomas Hofmeister. Depth efficient neural networks for division and related problems. *IEEE Trans. Inform. Theory*, 39(3):946–956, 1993. ISSN 0018-9448. doi: 10.1109/18.256501.

[80] Daniel Soudry and Elad Hoffer. Exponentially vanishing sub-optimal local minima in multilayer neural networks. *arXiv preprint arXiv:1702.05777*, 2017.

[81] Ryan Theisen, Jason Klusowski, and Michael Mahoney. Good classifiers are abundant in the interpolating regime. In *International Conference on Artificial Intelligence and Statistics*, pages 3376–3384. PMLR, 2021.

[82] Christos Thrampoulidis, Samet Oymak, and Mahdi Soltanolkotabi. Theoretical insights into multiclass classification: A high-dimensional asymptotic view. *Advances in Neural Information Processing Systems*, 33:8907–8920, 2020.

[83] Alexander Tsigler and Peter L. Bartlett. Benign overfitting in ridge regression. *Journal of Machine Learning Research*, 24(123):1–76, 2023.

[84] Guillermo Valle-Perez, Chico Q. Camargo, and Ard A. Louis. Deep learning generalizes because the parameter-function map is biased towards simple functions. In *International Conference on Learning Representations*, 2019.

[85] Gal Vardi, Gilad Yehudai, and Ohad Shamir. On the optimal memorization power of ReLU neural networks. In *International Conference on Learning Representations*, 2022.

[86] Roman Vershynin. Memory capacity of neural networks with threshold and relu activations. *arXiv preprint arXiv:2001.06938*, 2020.

[87] Guillaume Wang, Konstantin Donhauser, and Fanny Yang. Tight bounds for minimum l1-norm interpolation of noisy data. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2022.

[88] Ke Wang and Christos Thrampoulidis. Binary classification of gaussian mixtures: Abundance of support vectors, benign overfitting, and regularization. *SIAM Journal on Mathematics of Data Science*, 4(1):260–284, 2022.

[89] Ke Wang, Vidya Muthukumar, and Christos Thrampoulidis. Benign overfitting in multiclass classification: All roads lead to interpolation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[90] Denny Wu and Ji Xu. On the optimal weighted ℓ_2 regularization in overparameterized linear regression. *Advances in Neural Information Processing Systems*, 33:10112–10123, 2020.

[91] Xingyu Xu and Yuantao Gu. Benign overfitting of non-smooth neural networks beyond lazy training. In *International Conference on Artificial Intelligence and Statistics*, pages 11094–11117. PMLR, 2023.

[92] Zhiwei Xu, Yutong Wang, Spencer Frei, Gal Vardi, and Wei Hu. Benign overfitting and grokking in reLU networks for XOR cluster data. In *The Twelfth International Conference on Learning Representations*, 2024.

[93] Chulhee Yun, Suvrit Sra, and Ali Jadbabaie. Small relu networks are powerful memorizers: a tight analysis of memorization capacity. In *Advances in Neural Information Processing Systems*, pages 15558–15569, 2019.

[94] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3): 107–115, 2021.

[95] Lijia Zhou, Frederic Koehler, Pragya Sur, Danica J. Sutherland, and Nathan Srebro. A non-asymptotic moreau envelope theory for high-dimensional generalized linear models. In *Advances in Neural Information Processing Systems*, 2022.

[96] Lijia Zhou, James B Simon, Gal Vardi, and Nathan Srebro. An agnostic view on the cost of overfitting in (kernel) ridge regression. In *The Twelfth International Conference on Learning Representations*, 2024.

Appendix A. Supplementary material

The first section of the appendix consists of supplementary results and discussions. The rest contain the complete proofs of our results.

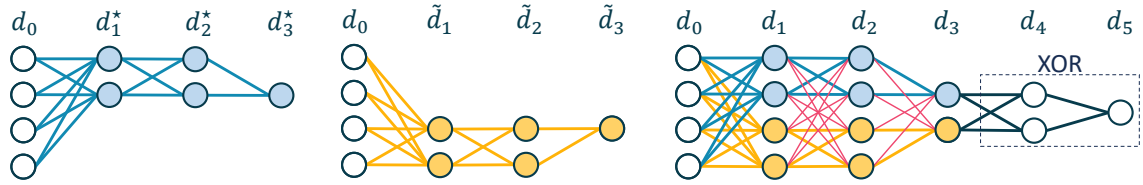
A.1. Supplementary material for Section 3

Proof idea for Theorem 4. We begin by noticing that under a teacher model $h^\star$ (Assumption 2), the labels of a consistent dataset S (Def. 3) can be decomposed as

$$\forall i \in [N] \;\; y_i = h^\star(\mathbf{x}_i) \oplus f(\mathbf{x}_i) \;,$$

where $f : \{0,1\}^{d_0} \to \{0,1\}$ indicates a label flip in the ith example (and can be defined arbitrarily for $\mathbf{x} \notin S$). Motivated by this observation, we seek an upper bound for the dimensions of a noise-memorizing network $\tilde{h}_S$ implementing f. To derive such a bound, we employ established techniques from the pseudorandomness literature to construct an efficient *hitting set generator* (HSG)[3] for the class of all conjunctions of literals. The HSG definition implies that there exists a seed on which the generator outputs a truth table that agrees with f on S. The network $\tilde{h}_S$ computes any requested bit of that truth table (see the proof of Theorem 46 for more details). We then show how a "student" network with these dimensions can implement the XOR between $h^\star$ and the noise memorizing f for all points in S. In Figure 2, we illustrate the teacher network from Assumption 2, the noise-memorizing network $\tilde{h}_S$, and a wider student network that interpolates the training set S by using the XOR construction (formalized in Lemma 50).

Remark 8 (Dependence on d_0) *In Appendix F we show that the $O\left(d_0^2 \cdot \log N\right)$ term is nearly tight, yet it can be relaxed when using some closely related NN architectures. For example, with a single additional layer of width $\Omega\left(\sqrt{d_0} \cdot \log N\right)$ with ternary weights in the first layer, i.e., $\mathcal{Q}_1^W = \{-1,0,1\}$ instead of $\{0,1\}$, the $O\left(d_0^2 \cdot \log N\right)$ term of Theorem 4 can be improved to $O\left(d_0^{3/2} \cdot \log N + d_0 \cdot \log^3 N\right)$.*



(a) A teacher model $h^\star$. (b) A NN $\tilde{h}_S$ memorizing the label flips. (c) A wider student NN can interpolate the training set S, *e.g.,* using a XOR construction.

Figure 2: **Interpolating a dataset.** To memorize the training set, we use a subset of the parameters to match those of the teacher and another subset to memorize the noise (label flips). Then, we "merge" these subsets to interpolate the noisy training set. In our figure, (1) blue edges represent weights identical to the teacher's; (2) yellow edges memorize the noise; (3) red edges are set to 0; and two additional layers implement the XOR between outputs, thus memorizing the training set.

3. A variant of the *pseudorandom generator* (PRG) concept.

A.2. Supplementary material for Section 4

The overfitting behaviors described in this section are illustrated in Figure 1.

Equivalent definition of posterior sampling. For consistent S, $\mathcal{P}_S$ can be equivalently defined as rejection sampling from the uniform distribution over $\Theta^{\text{BTN}}(\underline{d})$. That is, if we let $\mathcal{P}(h) = \mathbb{P}_{\boldsymbol{\theta}\sim\text{Uniform}(\Theta^{\text{BTN}}(\underline{d}))}(h_{\boldsymbol{\theta}} = h)$ then $\mathcal{P}_S(h) = \mathcal{P}(h \mid \mathcal{L}_S(h) = 0) = \mathbb{I}\{\mathcal{L}_S(h) = 0\} \cdot \mathcal{P}(h)/p_S$ where $p_S \triangleq \mathcal{P}(\mathcal{L}_S(h) = 0)$ is the interpolation probability.

Proof idea for Theorems 6 and 7. We extend the information-theoretical generalization bounds from [58] to this paper's setting in which label collisions in the datasets have a non-zero probability. In particular, we bound the interpolator's complexity from below by the mutual information between the model and the training set. Since the model is interpolating, we can further bound the mutual information by a quantity dependent on the population error. From the other direction, we bound the model's complexity from above by (1) its size in the min-size setting of Section 4.1, and (2) by the negative log interpolation probability for the posterior sampling of Section 4.2. Together with Corollary 4 we obtain the bounds above on the expected generalization error.

In Figure 2 we illustrated the construction of a memorizing network used to bound the complexity of the min-size interpolator. In the following Figure 3 we illustrate how the interpolation probability p_S can be bounded so as to induce a meaningful generalization bound.

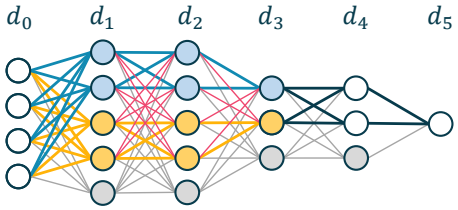


Figure 3: **Interpolating a dataset with an overparameterized student.** We build on the construction from Figure 2 that memorizes a dataset using a subset of the parameters (blue, yellow, and red edges). Then, redundant neurons (grey) can be effectively ignored by setting their neuron scaling parameters (γ) to 0, leaving the redundant weights (grey edges) unconstrained. Thus, the interpolation probability p_S can be bounded by a quantity exponentially decaying in the number of neurons $n(\underline{d})$ rather than in the number of weights $w(\underline{d}) = \omega(N)$.

Following Remark 8, the assumption $N = \omega\left(d_0^2 \log d_0\right)$ can be relaxed in some related architectures. For example, with a single additional layer of width $O\left(\sqrt{d_0} \cdot \log N\right)$ and ternary weights in the first layer $\mathcal{Q}_1^W = \{-1, 0, 1\}$, the requirement can be relaxed to $N = \omega\left(d_0^{3/2} \log d_0\right)$.

Remark 9 (Higher weight quantization) *The results bounds in the arbitrary noise setting can easily be extended to NNs with higher quantization levels. For example, letting $\mathcal{Q}_l^W$ such that $\left|\mathcal{Q}_l^W\right| = Q$ and $\{0, 1\} \subseteq \mathcal{Q}_l^W$, under the appropriate assumptions we get that*

$$\mathbb{E}_{(S,A(S))}\left[\mathcal{L}_{\mathcal{D}}(A(S))\right] \lessapprox 1 - Q^{-H(\varepsilon^\star)},$$

which is a meaningful bound for noise levels $\varepsilon^\star \leq \varepsilon(Q)$ for some $\varepsilon(Q) < 1/2$.[4] *Tighter results would require utilizing the additional quantization levels to achieve smaller dimensions of the interpolating network, and are left to future work.*

4. Specifically $\varepsilon(Q)$ such that $1 - Q^{-H(\varepsilon(Q))} \leq \frac{1}{2}$.

A.3. Related work

Benign and tempered overfitting. The benign overfitting phenomenon has been extensively studied in recent years. Previous works analyzed the conditions in which benign overfitting occurs in linear regression [3, 5, 11, 19, 21, 31, 34, 44, 47, 65, 67, 75, 83, 87, 90, 95], kernel regression [6, 8, 10, 50, 51, 53, 57, 60, 61, 71, 96], and linear classification [14, 18, 25, 52, 64, 66, 75, 82, 88, 89]. Moreover, several works proved benign overfitting in classification using nonlinear NNs [15, 28–30, 46, 48, 49, 62, 91, 92]. We note that all the aforementioned benign overfitting results require high-dimensional settings, namely, the input dimension is larger than the number of training samples.

Mallinar et al. [57] suggested the taxonomy of benign, tempered, and catastrophic overfitting, which we use in this work. They demonstrated empirically that nonlinear NNs in classification tasks exhibit tempered overfitting. Tempered overfitting in kernel ridge regression was theoretically studied in Barzilai and Shamir [6], Mallinar et al. [57], Zhou et al. [96]. In univariate ReLU NNs (namely, for input dimension 1), tempered overfitting was obtained for both classification [48] and regression [43]. Manoj and Srebro [58] proved tempered overfitting for a learning rule returning short programs in some programming language. Finally, tempered overfitting is well understood for the 1-nearest-neighbor learning rule, where the asymptotic risk is roughly twice the Bayes risk [23].

Circuit complexity. Theorem 46 (our NN for memorizing label flips) is in a similar spirit as several prior theorems in the area of *circuit complexity*. For example, Lupanov famously proved that every function $f\colon \{0,1\}^{d_0} \to \{0,1\}$ can be computed by a circuit consisting of $(1+o(1)) \cdot 2^{d_0}/d_0$ many AND/OR/NOT gates, where the AND/OR gates have fan-in two [55]. Lupanov's bound, which is tight [76], is analogous to Theorem 46, because a NN can be considered a type of circuit.

Even more relevant is a line of work that analyzes the circuit complexity of an arbitrary partial function $f\colon \{0,1\}^{d_0} \to \{0,1,\star\}$ with a given domain size N and a given number of 1-inputs N_1, similar to the setup of Theorem 46. See Jukna's textbook for an overview [45, Section 1.4.2]. We highlight the work of Chashkin, who showed that every such function can be computed by a circuit (of unbounded depth and bounded fan-in) with $(1+o(1)) \cdot \frac{\log \binom{N}{N_1}}{\log\log \binom{N}{N_1}} + O(d_0)$ gates [17].

To the best of our knowledge, prior to our work, nothing analogous to Chashkin's theorem [17] was known regarding constant-depth threshold networks. It is conceivable that one could adapt Chashkin's construction [17] to the binary threshold network setting as a method of proving Theorem 46, but our proof of Theorem 46 uses a different approach. Our proof relies on shallow threshold networks computing *k-wise independent generators* [36] and an *error-reduction* technique that was developed in the context of space-bounded derandomization [38], among other ingredients.

Memorization. Our construction shows how noisy data can be interpolated using a small threshold NN with binary weights. It essentially requires memorizing the noisy examples. The task of memorization, namely, finding a smallest NN that allows for interpolation of arbitrary data points, has been extensively studied in recent decades. Memorization of N arbitrary points in general position in $\mathbb{R}^d$ with a two-layer NN can be achieved using $O\left(\lceil \frac{N}{d} \rceil\right)$ hidden neurons [7, 12, 80]. Memorizing arbitrary N points, even if they are not in general position, can be done using two-layer networks with $O(N)$ neurons [40, 41, 73, 94]. With three-layer networks, $O(\sqrt{N})$ neurons suffice, but the number of parameters is still linear in N [39, 70, 86, 93]. Using deeper networks allows for memorization with a sublinear number of parameters [68, 85]. For example, memorization with networks of depth $\sqrt{N}$ requires only $\tilde{O}(\sqrt{N})$ parameters [85]. However, we note that in the aforementioned results, the number of quantization levels is not constant, namely, the number of bits

in the representation of each weight depends on N.[5]. Moreover, even in the sublinear constructions of [68, 85], the number of bits required to represent the network is $\omega(N)$. As a result, in this work we cannot rely on these constructions to obtain meaningful bounds.

Posterior sampling and guess and check. The generalization of random interpolating NNs has previously been studied, both empirically and theoretically [13, 20, 63, 81, 84]. Theisen et al. [81] studied the generalization of interpolating random linear and random features classifiers. Mingard et al. [63], Valle-Perez et al. [84] considered the Gaussian process approximation to random NNs which typically requires networks with infinite width. Buzaglo et al. [13] provided a method to obtain generalization results for quantized random NNs of general architectures — possibly deep and with finite width, under the assumption of a narrow teacher model. A variant of this approach was used to prove our generalization results of posterior sampling, with the XOR network (Lemma 50) used in the role of the teacher.

5. We note that in most papers, the required number of quantization levels is implicit in the constructions, and is not discussed explicitly

Appendix B. Preliminaries and Auxiliary Results

B.1. Preliminaries

Before moving to the proofs of the main results, we recall and introduce some notation that will be used throughout the supplementary material.

Notation. We denote a (possibly random) learning algorithm by $A(S)$. We use $\mathcal{D}$ to denote the joint distribution over a finite sample space $\mathcal{X} \times \{0,1\}$ of the features and labels, ν to denote the marginal distribution of the algorithm, and p to denote the joint distribution of a training set $S \sim \mathcal{D}^N$ and the algorithm $A(S)$. Specifically, the training set is a random element

$$S = \{(X_1, Y_1), \ldots, (X_N, Y_N)\} \sim \mathcal{D}^N$$

where (X_i, Y_i) is reserved for the i-th example in S. That is (X_i, Y_i) is always a sample in S, whereas (X, Y) is used to denote a data point which is independent of S. We use $d\mathcal{D}(x, y)$, $d\nu(h)$ and $dp(s, h) = dp(\{(x_1, y_1), \ldots, (x_N, y_N)\}, h)$ to denote the corresponding probability mass functions. With some abuse of notation, we use $d\mathcal{D}(x)$ for the probability mass function of the marginal of $\mathcal{D}$ over $\mathcal{X}$

$$d\mathcal{D}(x) = \mathbb{P}_{(X,Y)\sim\mathcal{D}}(X = x) ,$$

and $dp((x_1, y_1), h)$ for the marginal of the joint probability of a single point from S and the output of the algorithm, *i.e.,*

$$dp((x_1, y_1), h) = \mathbb{P}_{(S,A(S))\sim p}(X_1 = x_1, Y_1 = y_1, A(S) = h) .$$

Similarly, we use $dp(x_1, h)$, $dp(y_1 \mid x_1, h)$, etc., for the probability mass functions of the appropriate marginal and conditional distributions.

Interpolating algorithm. In order to simplify the analysis, we introduce a framework of interpolation learning related to the one introduced in Framework 1.

Let $\tilde{A}(S)$ be a learning rule satisfying Framework 1, and let $\star$ be some arbitrary token distinct from any hypothesis the algorithm may produce. We define a modified learning rule $A(S)$[6] such that

- If S is inconsistent then $A(S) = \star$.
- Otherwise, if S is consistent then $A(S) = \tilde{A}(S)$, so in particular $\mathcal{L}_S(A(S)) = 0$.

Notice that since the $A(S) = \tilde{A}(S)$ when S is consistent

$$\mathbb{E}\left[\mathcal{L}_{\mathcal{D}}(A(S)) \mid \text{consistent } S\right] = \mathbb{E}\left[\mathcal{L}_{\mathcal{D}}\left(\tilde{A}(S)\right) \mid \text{consistent } S\right]$$

and therefore we can find bounds for the generalization error of $\tilde{A}(S)$ by analyzing $A(S)$. In addition, when it can be inferred from context we use $A(S)$ to denote the min-size and posterior sampling interpolators (instead of $A_L(S)$ or $A_{\underline{d}}(S)$, respectively).

For ease of exposition, throughout the appendix, we rephrase the assumptions made in Section 4, namely, that $N = \omega\left(d_0^2 \log d_0\right)$ and $N = o\left(1/\sqrt{\mathcal{D}_{\max}}\right)$, as follows.

Assumption 10 (Bounded input dimension) $d_0 = o\left(\sqrt{N/\log N}\right)$.

Assumption 11 (Data distribution flatness) $\mathcal{D}_{\max} = o\left(1/N^2\right)$.

6. As most of the appendix deals with the modified learning rule, we use $\tilde{A}(S)$ for the original one and $A(S)$ for the modified one.

B.2. Auxiliary results

We start by citing several standard results from information theory and lemmas from Manoj and Srebro [58] which will be useful throughout our supplementary materials.

Lemma 12 (Chain rule of mutual information) *For any random variables A_1, A_2 and B*

$$I\left(\left(A_1, A_2\right); B\right) = I\left(A_2; B \mid A_1\right) + I\left(A_1; B\right) .$$

Lemma 13 *Let A and B be any two random variables with associated marginal distributions p_A, p_B, and joint $p_{A,B}$. Let $q_{A|B}$ be any conditional distribution (i.e. such that for any b, $q_{A|B}\left(\cdot, b\right)$ is a normalized non-negative measure). Then:*

$$I\left(A; B\right) \geq \mathbb{E}_{A,B\sim p_{A,B}}\left[\log\left(\frac{dq_{A|B}\left(A|B\right)}{dp_A\left(A\right)}\right)\right] .$$

Lemma 14 *Let A_1, A_2, B be random variables where A_1 and A_2 are independent. Then*

$$I\left(\left(A_1, A_2\right); B\right) \geq I\left(A_1; B\right) + I\left(A_2; B\right) .$$

Lemma 15 (Lemma A.4 from Manoj and Srebro [58]) *For $C \geq 0$ and $0 \leq \alpha \leq 1$ it holds that*

$$1 - 2^{-H(\alpha)-C} \leq 1 - 2^{-H(\alpha)} + C .$$

Lemma 16 *Let $\varepsilon \in \left(0, \frac{1}{2}\right)$ and*

$$\phi\left(t\right) \triangleq \phi_\varepsilon\left(t\right) = \frac{\varepsilon^t}{\varepsilon^t + \left(1-\varepsilon\right)^t} = \frac{1}{1 + \left(\frac{1}{\varepsilon} - 1\right)^t} .$$

Then, ϕ is monotonically decreasing as a function of t, and convex in $(0, \infty)$.

Proof Denote $\alpha \triangleq \frac{1}{\varepsilon} - 1$ then

$$\begin{aligned}
\phi\left(t\right) &= \frac{1}{1+\alpha^t} \\
\phi'\left(t\right) &= \frac{-\ln\left(\alpha\right)\alpha^t}{\left(1+\alpha^t\right)^2} = -\ln\left(\alpha\right) \cdot \frac{\alpha^t}{1 + 2\alpha^t + \alpha^{2t}} \\
\phi''\left(t\right) &= -\ln\left(\alpha\right) \cdot \frac{\ln\left(\alpha\right)\alpha^t\left(1+\alpha^t\right)^2 - \alpha^t \cdot 2\left(1+\alpha^t\right) \cdot \ln\left(\alpha\right)\alpha^t}{\left(1+\alpha^t\right)^4} \\
&= -\ln\left(\alpha\right)^2 \cdot \alpha^t \cdot \frac{\left(1+\alpha^t\right) - 2\alpha^t}{\left(1+\alpha^t\right)^3} = \ln\left(\alpha\right)^2 \cdot \alpha^t \cdot \frac{\alpha^t - 1}{\left(1+\alpha^t\right)^3} .
\end{aligned}$$

Notice that for any $\varepsilon \in \left(0, \frac{1}{2}\right)$, $\alpha = \frac{1}{\varepsilon} - 1 > 1$ so for all $t > 0$

$$\alpha^t - 1 > 0$$

and $\phi''\left(t\right) > 0$ so the function is indeed convex, and $-\ln\left(\alpha\right) < 0$ so ϕ is decreasing. ■

Corollary 17 *For all $t > 0$ it holds that*

$$\phi(t) \geq \phi(1) + \phi'(1)(t-1) = \varepsilon + \ln 2 \left(\varepsilon \log(\varepsilon) + \varepsilon H(\varepsilon)\right)(t-1) \ .$$

Proof Substituting $t = 1$,

$$\begin{aligned}
\phi(1) &= \frac{\varepsilon}{\varepsilon + (1-\varepsilon)} = \varepsilon \\
\phi'(1) &= -\ln(\alpha) \cdot \frac{\alpha}{(1+\alpha)^2} = -\ln\left(\frac{1}{\varepsilon} - 1\right) \cdot \frac{\frac{1}{\varepsilon} - 1}{\left(1 + \left(\frac{1}{\varepsilon} - 1\right)\right)^2} = -\ln\left(\frac{1-\varepsilon}{\varepsilon}\right) \cdot \frac{\frac{1}{\varepsilon} - 1}{\left(\frac{1}{\varepsilon}\right)^2} \\
&= -\left(\ln(1-\varepsilon) - \ln(\varepsilon)\right) \cdot \left(\varepsilon - \varepsilon^2\right) = \varepsilon(1-\varepsilon)\ln(\varepsilon) - \varepsilon(1-\varepsilon)\ln(1-\varepsilon) \\
&= \varepsilon \ln(\varepsilon) - \varepsilon\left(\varepsilon \ln(\varepsilon) + (1-\varepsilon)\ln(1-\varepsilon)\right) \\
&= \varepsilon \ln 2 \left(\log(\varepsilon) - \left(\varepsilon \log(\varepsilon) + (1-\varepsilon)\log(1-\varepsilon)\right)\right) \\
&= \varepsilon \ln 2 \left(\log(\varepsilon) + H(\varepsilon)\right) = \ln 2 \left(\varepsilon \log(\varepsilon) + \varepsilon H(\varepsilon)\right) \ .
\end{aligned}$$

The inequality then holds due to convexity. ■

Appendix C. Data consistency

Before moving on to generalization, we address some key properties of the training set's consistency.

Lemma 18 *For any distribution over the data $\mathcal{D}$, $\mathbb{P}\left(\textit{inconsistent } S\right) \leq \frac{1}{2}N^2\mathcal{D}_{\max}$.*

Proof Using the union bound,

$$\begin{aligned}
\mathbb{P}\left(\text{inconsistent } S\right) &= \mathbb{P}\left(\exists i \neq j \in [N] \,:\, X_i = X_j, Y_i \neq Y_j\right) \\
&\leq \mathbb{P}\left(\exists i \neq j \in [N] \,:\, X_i = X_j\right) \\
&\leq \sum_{i \neq j} \mathbb{P}\left(X_i = X_j\right) = \binom{N}{2}\mathbb{P}\left(X_1 = X_2\right) = \binom{N}{2}\sum_{x \in \mathcal{X}} \mathbb{P}\left(X_1 = x\right)\mathbb{P}\left(X_2 = x\right) \\
&\leq \binom{N}{2}\sum_{x \in \mathcal{X}} \mathcal{D}_{\max}\mathbb{P}\left(X = x\right) = \binom{N}{2}\mathcal{D}_{\max} \leq \frac{1}{2}N^2\mathcal{D}_{\max}\,.
\end{aligned}$$

■

Hence, under Assumption 11 we have $\mathbb{P}\left(\text{inconsistent } S\right) = o\left(1\right)$, *i.e.,* the inconsistency probability is asymptotically small.

C.1. Independent label noise

We now focus on the case of independent label noise, *i.e.,* $Y \oplus h^\star\left(X\right) \mid \{X = x\} \sim \text{Ber}\left(\varepsilon^\star\right)$ for any $x \in \mathcal{X}$. Recall the noise level

$$\varepsilon^\star = \mathbb{P}_{(X,Y)\sim\mathcal{D}}\left(Y \neq h^\star\left(X\right)\right) = \mathbb{P}_S\left(Y_1 \neq h^\star\left(X_1\right)\right)$$

and we define the "effective" noise level in a *consistent* training set

$$\hat{\varepsilon}_{\text{tr}} \triangleq \mathbb{P}_S\left(Y_1 \neq h^\star\left(X_1\right) \mid \text{consistent } S\right)\,. \tag{2}$$

We relate $\hat{\varepsilon}_{\text{tr}}$ to $\varepsilon^\star$ in the following lemma.

Lemma 19 *In the independent noise setting, it holds that*

$$|\hat{\varepsilon}_{tr} - \varepsilon^\star| \leq \left|\ln 2\left(\varepsilon^\star \log\left(\varepsilon^\star\right) + \varepsilon^\star H\left(\varepsilon^\star\right)\right)\right| \cdot \left(N-1\right)\frac{\mathcal{D}_{\max}}{\mathbb{P}\left(\text{consistent } S\right)}\,,$$

and moreover, $\hat{\varepsilon}_{tr} \leq \varepsilon^\star$.

Proof Conditioning on S being consistent (having no label "collisions"), all occurrences of x in S must have the same label so

$$\mathbb{P}_S\left(Y_1 \neq h^\star\left(X_1\right) \,|\, (X_1, Y_1) \text{ appears } k \text{ times in } S, \text{ consistent } S\right) = \frac{\varepsilon^{\star k}}{\varepsilon^{\star k} + \left(1 - \varepsilon^\star\right)^k}$$

Therefore,

$$\begin{aligned}
\hat{\varepsilon}_{\text{tr}} &= \mathbb{P}_S\left(Y_1 \neq h^\star\left(X_1\right) \mid \text{consistent } S\right) \\
&= \sum_{k=1}^{N} \mathbb{P}_S\left(Y_1 \neq h^\star\left(X_1\right) \mid (X_1, Y_1) \text{ appears } k \text{ times in } S, \text{ consistent } S\right) \\
&\qquad \cdot \mathbb{P}\left((X_1, Y_1) \text{ appears } k \text{ times in } S \mid \text{consistent } S\right) \\
&= \sum_{k=1}^{N} \frac{\varepsilon^{\star k}}{\varepsilon^{\star k} + \left(1-\varepsilon^\star\right)^k} \cdot \mathbb{P}\left((X_1, Y_1) \text{ appears } k \text{ times in } S \mid \text{consistent } S\right) \\
&\leq \sum_{k=1}^{N} \frac{\varepsilon^{\star 1}}{\varepsilon^{\star 1} + \left(1-\varepsilon^\star\right)^1} \cdot \mathbb{P}\left((X_1, Y_1) \text{ appears } k \text{ times in } S \mid \text{consistent } S\right) \\
&= \varepsilon^\star \underbrace{\sum_{k=1}^{N} \cdot \mathbb{P}\left((X_1, Y_1) \text{ appears } k \text{ times in } S \mid \text{consistent } S\right)}_{\text{sums to 1}} = \varepsilon^\star .
\end{aligned} \tag{3}$$

On the other hand, define

$$K\left(S\right) \triangleq \left|\left\{i \in [N] \mid X_i = X_1\right\}\right| = \sum_{i=1}^{N} \mathbb{I}\left\{X_i = X_1\right\}$$

then

$$\begin{aligned}
\mathbb{E}_S\left[K\left(S\right) \mid \text{consistent } S\right] &= 1 + \sum_{i=2}^{N} \mathbb{E}_S\left[\mathbb{I}\left\{X_i = X_1\right\} \mid \text{consistent } S\right] \\
&= 1 + \left(N-1\right) \mathbb{P}_S\left(X_2 = X_1 \mid \text{consistent } S\right) .
\end{aligned}$$

Next,

$$\mathbb{P}\left(X_2 = X_1 \mid \text{consistent } S\right) = \frac{\mathbb{P}\left(X_2 = X_1, d\mathcal{D}\left(\neg h^\star\left(X_1\right) \mid X_1\right) < \frac{1}{2}\right)}{\mathbb{P}\left(\text{consistent } S\right)} \leq \frac{\mathbb{P}\left(X_1 = X_2\right)}{\mathbb{P}\left(\text{consistent } S\right)} .$$

Since $d\mathcal{D}\left(x\right) \leq \mathcal{D}_{\max}$ for all $x \in \mathcal{X}$, as in the proof of Lemma 18

$$\mathbb{E}_S\left[K\left(S\right) \mid \text{consistent } S\right] \leq 1 + \left(N-1\right) \frac{\mathbb{P}\left(X_1 = X_2\right)}{\mathbb{P}\left(\text{consistent } S\right)} \leq 1 + \left(N-1\right) \frac{\mathcal{D}_{\max}}{\mathbb{P}\left(\text{consistent } S\right)} .$$

Then, using Lemma 16 we get,

$$\begin{aligned}
\hat{\varepsilon}_{\text{tr}} &= \sum_{k=1}^{N} \frac{\varepsilon^{\star k}}{\varepsilon^{\star k} + (1-\varepsilon^{\star})^{k}} \cdot \mathbb{P}\left(X_1 \text{ appears } k \text{ times in } S \mid \text{consistent } S\right) \\
&= \sum_{k=1}^{N} \phi_{\varepsilon^{\star}}(k) \cdot \mathbb{P}\left(X_1 \text{ appears } k \text{ times in } S \mid \text{consistent } S\right) \\
&= \mathbb{E}_S\left[\underbrace{\phi_{\varepsilon^{\star}}(K(S))}_{\text{convex in } k} \mid \text{consistent } S\right] \\
[\text{Jensen}] &\geq \phi_{\varepsilon^{\star}}\left(\mathbb{E}_S\left[K(S) \mid \text{consistent } S\right]\right) \\
[\text{decreasing}] &\geq \phi_{\varepsilon^{\star}}\left(1 + (N-1) \cdot \frac{\mathcal{D}_{\max}}{\mathbb{P}\left(\text{consistent } S\right)}\right).
\end{aligned}$$

Corollary 17 implies that

$$\begin{aligned}
\hat{\varepsilon}_{\text{tr}} &\geq \phi_{\varepsilon^{\star}}\left(1 + (N-1) \cdot \frac{\mathcal{D}_{\max}}{\mathbb{P}\left(\text{consistent } S\right)}\right) \\
&\geq \varepsilon^{\star} + \ln 2\left(\varepsilon^{\star}\log\left(\varepsilon^{\star}\right) + \varepsilon^{\star} H\left(\varepsilon^{\star}\right)\right) \cdot (N-1)\frac{\mathcal{D}_{\max}}{\mathbb{P}\left(\text{consistent } S\right)}.
\end{aligned}$$

Combining the bounds we get

$$\left|\hat{\varepsilon}_{\text{tr}} - \varepsilon^{\star}\right| \leq \left|\ln 2\left(\varepsilon^{\star}\log\left(\varepsilon^{\star}\right) + \varepsilon^{\star} H\left(\varepsilon^{\star}\right)\right)\right| \cdot (N-1)\frac{\mathcal{D}_{\max}}{\mathbb{P}\left(\text{consistent } S\right)}.$$

■

Appendix D. Generalization bounds

We present two generalization bounds for the population error of an interpolating algorithm in terms of the mutual information of it with the training set.

Remark 20 (High consistency probability) *Throughout the appendix we assume that* $\mathbb{P}_S\left(\text{consistent } S\right) \geq \frac{1}{2}$*. While this assumption is not without loss of generality, it is a weaker version of Assumption 11 and implied by it (asymptotically). As Assumption 11 is assumed in all "downstream results" that this appendix aims to support, we find it is reasonable to assume here.*

D.1. Arbitrary label noise

In this subsection, we provide a generalization bound for interpolating algorithms without any assumptions on the distribution of the noise $Y \oplus h^\star\left(X\right) \mid \{X = x\}$, other than $\mathcal{L}_{\mathcal{D}}\left(h^\star\right) = \varepsilon^\star$.

Lemma 21 *For any interpolating learning algorithm* $A\left(S\right)$*,*

$$-\log\left(1 - \mathbb{E}_{S,A(S)}\left[\mathcal{L}_{\mathcal{D}}\left(A\left(S\right)\right) \mid \text{consistent } S\right]\right) \leq \frac{I\left(S; A\left(S\right)\right)}{N \cdot \mathbb{P}_S\left(\text{consistent } S\right)} .$$

Proof We rely on Lemma 13. Specifically, we shall use the following suggested conditional distribution. For $h \neq \star$ let

$$dq\left(s|h\right) = \frac{1}{Z_h}\mathbb{I}\left\{\mathcal{L}_s\left(h\right) = 0\right\} d\mathcal{D}^N\left(s\right)$$

where

$$\begin{aligned} Z_h &= \sum_s \mathbb{I}\left\{\mathcal{L}_s\left(h\right) = 0\right\} d\mathcal{D}^N\left(s\right) \\ &= \mathbb{E}_S\mathbb{I}\left\{\mathcal{L}_S\left(h\right) = 0\right\} \\ &= \mathbb{P}_S\left(\mathcal{L}_S\left(h\right) = 0\right) \\ &= \left(1 - \mathcal{L}_{\mathcal{D}}\left(h\right)\right)^N . \end{aligned}$$

For $h = \star$ let

$$dq\left(s|\star\right) = \frac{\mathbb{I}\left\{\text{inconsistent } s\right\} d\mathcal{D}^N\left(s\right)}{\sum_{s'}\mathbb{I}\left\{\text{inconsistent } s'\right\} d\mathcal{D}^N\left(s'\right)} = \mathbb{I}\left\{\text{inconsistent } s\right\}\frac{d\mathcal{D}^N\left(s\right)}{\mathbb{P}_S\left(\text{inconsistent } S\right)} .$$

Clearly, if $h \neq \star$ and $dq\left(s|h\right) = 0$ then either $d\mathcal{D}^N\left(s\right) = 0$ so $dp\left(s, h\right) = 0$ as well, or $\mathcal{L}_s\left(h\right) \neq 0$. so, since $h \neq \star$, s can be interpolated and $dp\left(s \mid h\right) = 0$. That is, the proposed conditional distribution is absolutely continuous w.r.t. the true conditional distribution. When $h = \star$, q is the true conditional distribution given that $h = \star$ so it is also absolutely continuous w.r.t. it. That is, the proposed distribution is

$$dq\left(s \mid h\right) = \frac{\mathbb{I}\left\{\text{inconsistent } s\right\}}{\mathbb{P}_S\left(\text{inconsistent } S\right)}\mathbb{I}\left\{h = \star\right\} d\mathcal{D}^N\left(s\right) + \frac{\mathbb{I}\left\{\mathcal{L}_s\left(h\right) = 0\right\}}{\left(1 - \mathcal{L}_{\mathcal{D}}\left(h\right)\right)^N}\mathbb{I}\left\{h \neq \star\right\} d\mathcal{D}^N\left(s\right) .$$

From Lemma 13

$$\begin{aligned} I\left(S;A\left(S\right)\right) &\geq \mathbb{E}_{S,A(S)}\left[\log\left(\frac{dq\left(S|A\left(S\right)\right)}{d\mathcal{D}^{N}\left(S\right)}\right)\right] \\ &= \mathbb{E}_{S,A(S)}\left[\log\left(\frac{\mathbb{I}\{\text{inconsistent } S\}}{\mathbb{P}_{S'}(\text{inconsistent } S')}\mathbb{I}\left\{A\left(S\right)=\star\right\}+\frac{\mathbb{I}\{\mathcal{L}_{S}(A(S))=0\}}{\left(1-\mathcal{L}_{\mathcal{D}}(A(S))\right)^{N}}\mathbb{I}\left\{A\left(S\right)\neq\star\right\}\right)\right] . \end{aligned}$$

$\mathbb{I}\left\{A\left(S\right)=\star\right\}$ and $\mathbb{I}\left\{A\left(S\right)\neq\star\right\}$ are mutually exclusive so

$$\begin{aligned} &I\left(S;A\left(S\right)\right) \\ &\quad\geq \mathbb{E}_{S,A(S)}\left[\log\left(\frac{\mathbb{I}\{\text{inconsistent } S\}}{\mathbb{P}_{S'}(\text{inconsistent } S')}\right)\mathbb{I}\left\{A\left(S\right)=\star\right\}+\log\left(\frac{\mathbb{I}\{\mathcal{L}_{S}(A(S))=0\}}{\left(1-\mathcal{L}_{\mathcal{D}}(A(S))\right)^{N}}\right)\mathbb{I}\left\{A\left(S\right)\neq\star\right\}\right] . \end{aligned}$$

The first term is 0 when $A\left(S\right)\neq\star$ and positive when $A\left(S\right)=\star$ (and so always non-negative). Furthermore, since $dp\left(S,A\left(S\right)\right)>0$ and $\mathbb{I}\left\{A\left(S\right)\neq\star\right\}=1$ together imply that $\mathbb{I}\left\{\mathcal{L}_{S}\left(A\left(S\right)\right)=0\right\}=1$ we have

$$\begin{aligned} I\left(S;A\left(S\right)\right) &\geq \mathbb{E}_{S,A(S)}\left[\log\left(\frac{1}{\left(1-\mathcal{L}_{\mathcal{D}}\left(A\left(S\right)\right)\right)^{N}}\right)\mathbb{I}\left\{A\left(S\right)\neq\star\right\}\right] \\ &= -\mathbb{E}_{S,A(S)}\left[N\log\left(1-\mathcal{L}_{\mathcal{D}}\left(A\left(S\right)\right)\right)\mathbb{I}\left\{A\left(S\right)\neq\star\right\}\right] . \end{aligned}$$

Using Jensen's inequality,

$$\begin{aligned} &-\mathbb{E}_{S,A(S)}\left[N\log\left(1-\mathcal{L}_{\mathcal{D}}\left(A\left(S\right)\right)\right)\mathbb{I}\left\{A\left(S\right)\neq\star\right\}\right] \\ &\quad= -N\mathbb{E}_{S,A(S)}\left[\log\left(1-\mathcal{L}_{\mathcal{D}}\left(A\left(S\right)\right)\right)\mid\mathbb{I}\left\{A\left(S\right)\neq\star\right\}\right]\mathbb{P}_{S,A(S)}\left(A\left(S\right)\neq\star\right) \\ &\quad\geq -N\log\left(1-\mathbb{E}_{S,A(S)}\left[\mathcal{L}_{\mathcal{D}}\left(A\left(S\right)\right)\mid\mathbb{I}\left\{A\left(S\right)\neq\star\right\}\right]\right)\mathbb{P}_{S,A(S)}\left(A\left(S\right)\neq\star\right) \\ &\quad= -N\log\left(1-\mathbb{E}_{S,A(S)}\left[\mathcal{L}_{\mathcal{D}}\left(A\left(S\right)\right)\mid\text{consistent } S\right]\right)\mathbb{P}_{S}\left(\text{consistent } S\right) \end{aligned}$$

so

$$I\left(S;A\left(S\right)\right)\geq -N\log\left(1-\mathbb{E}_{S,A(S)}\left[\mathcal{L}_{\mathcal{D}}\left(A\left(S\right)\right)\mid\text{consistent } S\right]\right)\mathbb{P}_{S}\left(\text{consistent } S\right) .$$

Rearranging the inequality

$$-\log\left(1-\mathbb{E}_{S,A(S)}\left[\mathcal{L}_{\mathcal{D}}\left(A\left(S\right)\right)\mid\text{consistent } S\right]\right)\leq\frac{I\left(S;A\left(S\right)\right)}{N\cdot\mathbb{P}_{S}\left(\text{consistent } S\right)}$$

■

D.2. Independent Noise

Lemma 22 *Assuming independent noise, the generalization error of interpolating learning rules satisfies the following.*

$$\left|\mathbb{E}_{S,A(S)}\left[\mathcal{L}_{\mathcal{D}}\left(A\left(S\right)\right) \mid \text{consistent } S\right] - 2\varepsilon^{\star}\left(1-\varepsilon^{\star}\right)\right| \leq \left(1-2\varepsilon^{\star}\right)\sqrt{C\left(N\right)} + \frac{\left(N-1\right)\mathcal{D}_{\max}}{3},$$

where $C\left(N\right) \triangleq \frac{I(S;A(S))-N\cdot(H(\varepsilon^{\star})-\mathbb{P}(\text{inconsistent } S))}{N(1-\mathbb{P}(\text{inconsistent } S))}$.

Proof As in the proof of Lemma 4.2 in Manoj and Srebro [58], since S is sampled i.i.d., we have

$$\begin{aligned} I\left(S;A\left(S\right)\right) &\overset{14}{\geq} \sum_{i=1}^{N} I\left(\left(X_i,Y_i\right);A\left(S\right)\right) = N\cdot I\left(\left(X_1,Y_1\right);A\left(S\right)\right) \\ &\overset{12}{=} N\cdot I\left(X_1;A\left(S\right)\right) + N\cdot I\left(Y_1;A\left(S\right)\mid X_1\right). \end{aligned} \tag{4}$$

Using properties of conditional mutual information,

$$I\left(Y_1;A\left(S\right)\mid X_1\right) = H\left(Y_1\mid X_1\right) - H\left(Y_1\mid A\left(S\right),X_1\right). \tag{5}$$

For the first term in (5), we employ the fact that for any $x\in\mathcal{X}$, either $Y_1\mid X_1=x\sim\text{Ber}\left(\varepsilon^{\star}\right)$ or $Y_1\mid X_1=x\sim\text{Ber}\left(1-\varepsilon^{\star}\right)$ to get

$$H\left(Y_1\mid X_1\right) = \mathbb{E}_{(X,Y)\sim\mathcal{D}}\left[H\left(Y_1\mid X_1=X\right)\right] = \mathbb{E}_{(X,Y)\sim\mathcal{D}}\left[H\left(\varepsilon^{\star}\right)\right] = H\left(\varepsilon^{\star}\right).$$

For the second term in (5), we again employ the definition of conditional entropy,

$$\begin{aligned} H\left(Y_1\mid A\left(S\right),X_1\right) = -\sum_{x_1\in\mathcal{X}}\sum_{h\in\mathcal{H}\cup\{\star\}}\Bigg[&dp\left(\left(x_1,0\right),h\right)\log\left(\frac{dp\left(\left(x_1,0\right),h\right)}{dp\left(x_1,h\right)}\right) \\ &+ dp\left(\left(x_1,1\right),h\right)\log\left(\frac{dp\left(\left(x_1,1\right),h\right)}{dp\left(x_1,h\right)}\right)\Bigg]. \end{aligned}$$

When $h\neq\star$, the marginal distribution of a training datapoint and a hypothesis is

$$dp\left(\left(x,y\right),h\right) = dp\left(y\mid x,h\right)dp\left(x,h\right) = \mathbb{I}\left\{y=h\left(x\right)\right\}dp\left(x,h\right),$$

and the inner expression becomes:

$$dp\left(\left(x_1,h\left(x_1\right)\right),h\right)\log\left(\frac{dp\left(\left(x_1,h\left(x_1\right)\right),h\right)}{dp\left(x_1,h\right)}\right) = dp\left(x_1,h\right)\underbrace{\log\left(\frac{dp\left(x_1,h\right)}{dp\left(x_1,h\right)}\right)}_{=0} = 0.$$

Therefore, we have that,

$$\begin{aligned} &H\left(Y_1\mid A\left(S\right),X_1\right) \\ &= -\sum_{x_1\in\mathcal{X}}\left[dp\left(\left(x_1,0\right),\star\right)\log\left(\frac{dp\left(\left(x_1,0\right),\star\right)}{dp\left(x_1,\star\right)}\right) + dp\left(\left(x_1,1\right),\star\right)\log\left(\frac{dp\left(\left(x_1,1\right),\star\right)}{dp\left(x_1,\star\right)}\right)\right] \end{aligned}$$

Employing conditional probabilities (notice that $\frac{dp((x_1,0),\star)}{dp(x_1,\star)} = \frac{dp(0|x_1,\star)dp(x_1,\star)}{dp(x_1,\star)} = dp\left(0 \mid x_1,\star\right)$), we get,

$$\begin{aligned}
&H\left(Y_1 \mid A\left(S\right), X_1\right) \\
&= -\sum_{x_1\in\mathcal{X}} dp\left(x_1,\star\right)\left[dp\left(0 \mid x_1,\star\right)\log\left(dp\left(0 \mid x_1,\star\right)\right) + dp\left(1 \mid x_1,\star\right)\log\left(dp\left(1 \mid x_1,\star\right)\right)\right] \\
&= \sum_{x_1\in\mathcal{X}} dp\left(x_1,\star\right) H\left(dp\left(0 \mid x_1,\star\right)\right) = dp\left(\star\right)\sum_{x_1\in\mathcal{X}} dp\left(x_1 \mid \star\right) H\left(dp\left(0 \mid x_1,\star\right)\right) \\
&= \mathbb{P}\left(A\left(S\right)=\star\right)\mathbb{E}_{(S,A(S))\sim p}\Big[\underbrace{H\left(dp\left(0 \mid X_1,\star\right)\right)}_{\leq 1} \mid \text{inconsistent } S\Big] \leq \mathbb{P}_S\left(\text{inconsistent } S\right) .
\end{aligned}$$

Overall, the right term in (4) is lower bounded by,

$$I\left(Y_1; A\left(S\right) \mid X_1\right) = H\left(Y_1 \mid X_1\right) - H\left(Y_1 \mid A\left(S\right), X_1\right) \geq H\left(\varepsilon^\star\right) - \mathbb{P}\left(\text{inconsistent } S\right) .$$

For the left term, *i.e.,* $I\left(X_1; A\left(S\right)\right)$, we use the variational bound Lemma 13 with the following suggested conditional distribution.

- For $h=\star$, choose $dq\left(x_1 \mid \star\right) = dp\left(x_1\right)$ (notice that $\sum_{x_1\in\mathcal{X}} dq\left(x_1 \mid \star\right) = \sum_{x_1} dp\left(x_1\right) = 1$.
- Otherwise, if $h \neq \star$, denote $q_\varepsilon = \mathrm{Ber}\left(\varepsilon\right)$, and

$$\begin{aligned}
\hat{\varepsilon}_{\text{tr}} &= \mathbb{P}_S\left(Y_1 \neq h^\star\left(X_1\right) \mid \text{consistent } S\right) \\
\hat{\varepsilon}_{\text{gen}} &= \mathbb{E}_{(S,A(S))\sim p}\left[\mathbb{P}_{X\sim\mathcal{D}}\left(A\left(S\right)\left(X\right) \neq h^\star\left(X\right)\right) \mid A\left(S\right) \neq \star\right] .
\end{aligned}$$

Note that $\hat{\varepsilon}_{\text{tr}}$ may differ than $\varepsilon^\star$. We choose the following conditional distribution

$$dq\left(x_1 \mid h\right) = \frac{1}{Z_h}\cdot\frac{dq_{\hat{\varepsilon}_{\text{tr}}}\left(h\left(x_1\right)\oplus h^\star\left(x_1\right)\right)}{dq_{\hat{\varepsilon}_{\text{gen}}}\left(h\left(x_1\right)\oplus h^\star\left(x_1\right)\right)} dp\left(x_1\right) .$$

In total, we choose,

$$dq\left(x_1 \mid h\right) = \frac{1}{Z_h}\cdot\frac{dq_{\hat{\varepsilon}_{\text{tr}}}\left(h\left(x_1\right)\oplus h^\star\left(x_1\right)\right)}{dq_{\hat{\varepsilon}_{\text{gen}}}\left(h\left(x_1\right)\oplus h^\star\left(x_1\right)\right)}\cdot\mathbb{I}\left\{h\neq\star\right\} dp\left(x_1\right) + \mathbb{I}\left\{h=\star\right\} dp\left(x_1\right) ,$$

where Z_h is the corresponding partition function.

Then, we use (13) and properties of logarithms and indicators to show that,

$$
\begin{aligned}
I\left(X_1; A\left(S\right)\right) &\geq \mathbb{E}_{S,A(S)}\left[\log\left(\frac{dq\left(X_1|A\left(S\right)\right)}{dp\left(X_1\right)}\right)\right] \\
&= \mathbb{E}_{S,A(S)}\left[\log\left(\frac{\frac{1}{Z_{A(S)}}\frac{dq_{\hat{\varepsilon}_{\text{tr}}}(h(X_1)\oplus h^\star(X_1))}{dq_{\hat{\varepsilon}_{\text{gen}}}(h(X_1)\oplus h^\star(X_1))}dp\left(X_1\right)\mathbb{I}\left\{A\left(S\right)\neq\star\right\}+dp\left(X_1\right)\mathbb{I}\left\{A\left(S\right)=\star\right\}}{dp\left(X_1\right)}\right)\right] \\
&= \mathbb{E}_{S,A(S)}\left[\log\left(\frac{1}{Z_{A(S)}}\frac{dq_{\hat{\varepsilon}_{\text{tr}}}\left(h\left(X_1\right)\oplus h^\star\left(X_1\right)\right)}{dq_{\hat{\varepsilon}_{\text{gen}}}\left(h\left(X_1\right)\oplus h^\star\left(X_1\right)\right)}\mathbb{I}\left\{A\left(S\right)\neq\star\right\}+\mathbb{I}\left\{A\left(S\right)=\star\right\}\right)\right] \\
&= \mathbb{E}_{S,A(S)}\left[\log\left(\frac{1}{Z_{A(S)}}\frac{dq_{\hat{\varepsilon}_{\text{tr}}}\left(h\left(X_1\right)\oplus h^\star\left(X_1\right)\right)}{dq_{\hat{\varepsilon}_{\text{gen}}}\left(h\left(X_1\right)\oplus h^\star\left(X_1\right)\right)}\right)\mathbb{I}\left\{A\left(S\right)\neq\star\right\}\right]+ \\
&\quad \mathbb{E}_{S,A(S)}\Big[\underbrace{\log\left(1\right)\mathbb{I}\left\{A\left(S\right)=\star\right\}}_{=0}\Big] \\
&= \mathbb{E}_{S,A(S)}\left[\log\left(\frac{1}{Z_{A(S)}}\frac{dq_{\hat{\varepsilon}_{\text{tr}}}\left(h\left(X_1\right)\oplus h^\star\left(X_1\right)\right)}{dq_{\hat{\varepsilon}_{\text{gen}}}\left(h\left(X_1\right)\oplus h^\star\left(X_1\right)\right)}\right)\mathbb{I}\left\{A\left(S\right)\neq\star\right\}\right].
\end{aligned}
$$

Using the law of total expectation, the above becomes,

$$
= \mathbb{P}\left(A\left(S\right)\neq\star\right)\mathbb{E}_{S,A(S)}\left[\log\left(\frac{1}{Z_{A(S)}}\cdot\frac{dq_{\hat{\varepsilon}_{\text{tr}}}\left(h\left(X_1\right)\oplus h^\star\left(X_1\right)\right)}{dq_{\hat{\varepsilon}_{\text{gen}}}\left(h\left(X_1\right)\oplus h^\star\left(X_1\right)\right)}\right)\,\middle|\,A\left(S\right)\neq\star\right],
$$

where we also use Jensen's inequality to show,

$$
\begin{aligned}
&\mathbb{E}_{S,A(S)}\left[\log\left(\frac{1}{Z_{A(S)}}\cdot\frac{dq_{\hat{\varepsilon}_{\text{tr}}}\left(h\left(X_1\right)\oplus h^\star\left(X_1\right)\right)}{dq_{\hat{\varepsilon}_{\text{gen}}}\left(h\left(X_1\right)\oplus h^\star\left(X_1\right)\right)}\right)\,\middle|\,A\left(S\right)\neq\star\right] \\
&= \mathbb{E}_{S,A(S)}\left[\log\left(\frac{dq_{\hat{\varepsilon}_{\text{tr}}}\left(h\left(X_1\right)\oplus h^\star\left(X_1\right)\right)}{dq_{\hat{\varepsilon}_{\text{gen}}}\left(h\left(X_1\right)\oplus h^\star\left(X_1\right)\right)}\right)\,\middle|\,A\left(S\right)\neq\star\right]-\mathbb{E}_{S,A(S)}\left[\log\left(Z_{A(S)}\right)\mid A\left(S\right)\neq\star\right] \\
&\geq \mathbb{E}_{S,A(S)}\left[\log\left(\frac{dq_{\hat{\varepsilon}_{\text{tr}}}\left(h\left(X_1\right)\oplus h^\star\left(X_1\right)\right)}{dq_{\hat{\varepsilon}_{\text{gen}}}\left(h\left(X_1\right)\oplus h^\star\left(X_1\right)\right)}\right)\,\middle|\,A\left(S\right)\neq\star\right]-\log\left(\mathbb{E}_{S,A(S)}\left[Z_{A(S)}\mid A\left(S\right)\neq\star\right]\right).
\end{aligned}
$$

The partition function satisfies for all $h\neq\star$,

$$
\begin{aligned}
Z_h &= \sum_{x_1\in\mathcal{X}}\frac{dq_{\hat{\varepsilon}_{\text{tr}}}\left(h\left(x_1\right)\oplus h^\star\left(x_1\right)\right)}{dq_{\hat{\varepsilon}_{\text{gen}}}\left(h\left(x_1\right)\oplus h^\star\left(x_1\right)\right)}dp\left(x_1\right)=\mathbb{E}_{X\sim\mathcal{D}}\left[\frac{dq_{\hat{\varepsilon}_{\text{tr}}}\left(h\left(X\right)\oplus h^\star\left(X\right)\right)}{dq_{\hat{\varepsilon}_{\text{gen}}}\left(h\left(X\right)\oplus h^\star\left(X\right)\right)}\right] \\
&= \mathbb{P}_{X\sim\mathcal{D}}\left(h\left(X\right)=h^\star\left(X\right)\right)\cdot\frac{1-\hat{\varepsilon}_{\text{tr}}}{1-\hat{\varepsilon}_{\text{gen}}}+\mathbb{P}_{X\sim\mathcal{D}}\left(h\left(X\right)\neq h^\star\left(X\right)\right)\cdot\frac{\hat{\varepsilon}_{\text{tr}}}{\hat{\varepsilon}_{\text{gen}}}.
\end{aligned}
$$

Taking the expectation w.r.t. $(S, A\left(S\right))\sim p$, we get

$$
\begin{aligned}
&\mathbb{E}_{(S,A(S))\sim p}\left[Z_{A(S)}\mid A\left(S\right)\neq\star\right] \\
&= \frac{1-\hat{\varepsilon}_{\text{tr}}}{1-\hat{\varepsilon}_{\text{gen}}}\cdot\underbrace{\mathbb{E}_{S,A(S)}\left[\mathbb{P}\left(A\left(S\right)\left(X\right)=h^\star\left(X\right)\right)\mid A\left(S\right)\neq\star\right]}_{=1-\hat{\varepsilon}_{\text{gen}}}+ \\
&\qquad\qquad + \frac{\hat{\varepsilon}_{\text{tr}}}{\hat{\varepsilon}_{\text{gen}}}\cdot\underbrace{\mathbb{E}_{S,A(S)}\left[\mathbb{P}\left(A\left(S\right)\left(X\right)\neq h^\star\left(X\right)\right)\mid A\left(S\right)\neq\star\right]}_{=\hat{\varepsilon}_{\text{gen}}}=1.
\end{aligned}
$$

Combining the above, we have that,

$$I\left(X_1; A\left(S\right)\right) \geq \mathbb{P}\left(A\left(S\right) \neq \star\right) \mathbb{E}_{S,A(S)}\left[\log\left(\frac{dq_{\hat{\varepsilon}_{\text{tr}}}\left(A\left(S\right)\left(X_1\right) \oplus h^\star\left(X_1\right)\right)}{dq_{\hat{\varepsilon}_{\text{gen}}}\left(A\left(S\right)\left(X_1\right) \oplus h^\star\left(X_1\right)\right)}\right) \mid A\left(S\right) \neq \star\right].$$

Notice that $\left(A\left(S\right)\left(X_1\right) \oplus h^\star\left(X_1\right) \mid \{A\left(S\right) \neq \star\}\right) = \left(Y_1 \oplus h^\star\left(X_1\right) \mid \{\text{consistent } S\}\right)$, so $A\left(S\right)\left(X_1\right) \oplus h^\star\left(X_1\right) \mid \{A\left(S\right) \neq \star\} \sim \text{Ber}\left(\hat{\varepsilon}_{\text{tr}}\right)$, and thus

$$I\left(X_1; A\left(S\right)\right) \geq \mathbb{P}\left(A\left(S\right) \neq \star\right) D_{KL}\left(q_{\hat{\varepsilon}_{\text{tr}}} || q_{\hat{\varepsilon}_{\text{gen}}}\right) = \big(1 - \mathbb{P}\left(\text{inconsistent } S\right)\big) D_{KL}\left(q_{\hat{\varepsilon}_{\text{tr}}} || q_{\hat{\varepsilon}_{\text{gen}}}\right).$$

Putting this all together, (4) is lower bounded by,

$$I\left(S; A\left(S\right)\right) \geq N\big(1 - \mathbb{P}\left(\text{inconsistent } S\right)\big) D_{KL}\left(q_{\hat{\varepsilon}_{\text{tr}}} || q_{\hat{\varepsilon}_{\text{gen}}}\right) + N\big(H\left(\varepsilon^\star\right) - \mathbb{P}\left(\text{inconsistent } S\right)\big).$$

Rearranging the inequality

$$D_{KL}\left(q_{\hat{\varepsilon}_{\text{tr}}} || q_{\hat{\varepsilon}_{\text{gen}}}\right) \leq \frac{I\left(S; A\left(S\right)\right) - N \cdot \left(H\left(\varepsilon^\star\right) - \mathbb{P}\left(\text{inconsistent } S\right)\right)}{N\left(1 - \mathbb{P}\left(\text{inconsistent } S\right)\right)} \triangleq C\left(N\right).$$

Using Pinsker's inequality, we have,

$$\left|\hat{\varepsilon}_{\text{tr}} - \hat{\varepsilon}_{\text{gen}}\right| \leq \sqrt{\frac{1}{2} D_{KL}\left(q_{\hat{\varepsilon}_{\text{tr}}} || q_{\hat{\varepsilon}_{\text{gen}}}\right)} \leq \sqrt{C\left(N\right)}.$$

We proceed to bound $\mathbb{E}_{S,A(S)}\left[\mathcal{L}_{\mathcal{D}}\left(A\left(S\right)\right) \mid \text{consistent } S\right]$ in terms of $\left|\hat{\varepsilon}_{\text{tr}} - \hat{\varepsilon}_{\text{gen}}\right|$. Notice that

$$\begin{aligned}
&\mathbb{E}_{S,A(S)}\left[\mathcal{L}_{\mathcal{D}}\left(A\left(S\right)\right) \mid \text{consistent } S\right] \\
&= \mathbb{E}_{S,A(S)}\left[\mathbb{P}_{(X,Y)\sim\mathcal{D}}\left(A\left(S\right)\left(X\right) \neq Y\right) \mid \text{consistent } S\right] \\
&= \mathbb{E}_{S,A(S)}\Big[\mathbb{P}\left(A\left(S\right)\left(X\right) \neq Y \mid Y = h^\star\left(X\right)\right) \underbrace{\mathbb{P}\left(Y = h^\star\left(X\right)\right)}_{\text{no label flip}} \Big| \text{ consistent } S\Big] + \\
&\quad + \mathbb{E}_{S,A(S)}\Big[\mathbb{P}\left(A\left(S\right)\left(X\right) \neq Y \mid Y \neq h^\star\left(X\right)\right) \underbrace{\mathbb{P}\left(Y \neq h^\star\left(X\right)\right)}_{\text{label flip}} \Big| \text{ consistent } S\Big] \\
&= \mathbb{E}_{S,A(S)}\Big[\mathbb{P}\left(A\left(S\right)\left(X\right) \neq h^\star\left(X\right)\right)\left(1 - \varepsilon^\star\right) + \mathbb{P}\left(A\left(S\right)\left(X\right) = h^\star\left(X\right)\right)\varepsilon^\star \Big| \text{ consistent } S\Big] \\
&= \left(1 - \varepsilon^\star\right) \mathbb{E}_{S,A(S)}\left[\mathbb{P}\left(A\left(S\right)\left(X\right) \neq h^\star\left(X\right)\right) \mid \text{consistent } S\right] + \\
&\quad + \varepsilon^\star \mathbb{E}_{S,A(S)}\left[\mathbb{P}\left(A\left(S\right)\left(X\right) = h^\star\left(X\right)\right) \mid \text{consistent } S\right] \\
&= \left(1 - \varepsilon^\star\right)\hat{\varepsilon}_{\text{gen}} + \varepsilon^\star\left(1 - \hat{\varepsilon}_{\text{gen}}\right).
\end{aligned}$$

Then, using the triangle inequality,

$$\begin{aligned}
&\left|\mathbb{E}_{S,A(S)}\left[\mathcal{L}_{\mathcal{D}}\left(A\left(S\right)\right) \mid \text{consistent } S\right] - 2\varepsilon^\star\left(1 - \varepsilon^\star\right)\right| \\
&= \left|\left(1 - \varepsilon^\star\right)\hat{\varepsilon}_{\text{gen}} + \varepsilon^\star\left(1 - \hat{\varepsilon}_{\text{gen}}\right) - 2\varepsilon^\star\left(1 - \varepsilon^\star\right)\right| = \left|\hat{\varepsilon}_{\text{gen}} - \hat{\varepsilon}_{\text{gen}}\varepsilon^\star + \varepsilon^\star - \hat{\varepsilon}_{\text{gen}}\varepsilon^\star - 2\varepsilon^\star + 2\varepsilon^{\star 2}\right| \\
&= \left|\hat{\varepsilon}_{\text{gen}} - 2\hat{\varepsilon}_{\text{gen}}\varepsilon^\star - \varepsilon^\star + 2\varepsilon^{\star 2}\right| = \left|\hat{\varepsilon}_{\text{gen}}\left(1 - 2\varepsilon^\star\right) - \varepsilon^\star\left(1 - 2\varepsilon^\star\right)\right| = \left(1 - 2\varepsilon^\star\right)\left|\hat{\varepsilon}_{\text{gen}} - \varepsilon^\star\right| \\
&\leq \left(1 - 2\varepsilon^\star\right)\left(\left|\hat{\varepsilon}_{\text{gen}} - \hat{\varepsilon}_{\text{tr}}\right| + \left|\hat{\varepsilon}_{\text{tr}} - \varepsilon^\star\right|\right).
\end{aligned}$$

Combining with the result from Lemma 19 and Remark 20

$$\begin{aligned}|\hat{\varepsilon}_{\text{tr}} - \varepsilon^\star| &\leq |\ln 2 \left(\varepsilon^\star \log\left(\varepsilon^\star\right) + \varepsilon^\star H\left(\varepsilon^\star\right)\right)| \cdot (N-1) \frac{\mathcal{D}_{\max}}{\mathbb{P}\left(\text{consistent } S\right)} \\ &\leq |\ln 2 \left(\varepsilon^\star \log\left(\varepsilon^\star\right) + \varepsilon^\star H\left(\varepsilon^\star\right)\right)| \cdot 2 (N-1) \mathcal{D}_{\max} .\end{aligned}$$

we conclude that

$$\begin{aligned}&\left|\mathbb{E}_{S,A(S)}\left[\mathcal{L}_{\mathcal{D}}\left(A\left(S\right)\right) \mid \text{consistent } S\right] - 2\varepsilon^\star\left(1-\varepsilon^\star\right)\right| \\ &\leq \left(1-2\varepsilon^\star\right)\left(\sqrt{C\left(N\right)} + \ln 2 \left|\left(\varepsilon^\star \log\left(\varepsilon^\star\right) + \varepsilon^\star H\left(\varepsilon^\star\right)\right)\right| \cdot 2\left(N-1\right)\mathcal{D}_{\max}\right) .\end{aligned}$$

Finally, we can use the algebraic property that $\left(1-2\varepsilon^\star\right)\left|\ln 2 \left(\varepsilon^\star \log\left(\varepsilon^\star\right) + \varepsilon^\star H\left(\varepsilon^\star\right)\right)\right| \leq \frac{1}{6}$, to get

$$\begin{aligned}&\left|\mathbb{E}_{S,A(S)}\left[\mathcal{L}_{\mathcal{D}}\left(A\left(S\right)\right) \mid \text{consistent } S\right] - 2\varepsilon^\star\left(1-\varepsilon^\star\right)\right| \\ &\qquad\qquad \leq \left(1-2\varepsilon^\star\right)\sqrt{C\left(N\right)} + \frac{\left(N-1\right)\mathcal{D}_{\max}}{3} .\end{aligned}$$

■

We can now bound the expected generalization error without conditioning on the consistency of the training set.

Lemma 23 *It holds that,*

$$\begin{aligned}&\left|\mathbb{E}_{S,A(S)}\left[\mathcal{L}_{\mathcal{D}}\left(A\left(S\right)\right)\right] - 2\varepsilon^\star\left(1-\varepsilon^\star\right)\right| \\ &\qquad \leq \left|\mathbb{E}_{S,A(S)}\left[\mathcal{L}_{\mathcal{D}}\left(A\left(S\right)\right) \mid \text{consistent } S\right] - 2\varepsilon^\star\left(1-\varepsilon^\star\right)\right| + \mathbb{P}_S\left(\text{inconsistent } S\right) .\end{aligned}$$

Proof Let X be an arbitrary RV in $[0,1]$ and Y be a binary RV. Then,

$$\begin{aligned}\mathbb{E}\left[X\right] &= \mathbb{P}\left(Y\right)\mathbb{E}\left[X|Y\right] + \mathbb{P}\left(\neg Y\right)\mathbb{E}\left[X|\neg Y\right] \\ \mathbb{E}\left[X\right] - \mathbb{E}\left[X|Y\right] &= \mathbb{P}\left(Y\right)\mathbb{E}\left[X|Y\right] - \mathbb{E}\left[X \mid Y\right] + \mathbb{P}\left(\neg Y\right)\mathbb{E}\left[X|\neg Y\right] \\ &= \mathbb{E}\left[X|Y\right]\left(\mathbb{P}\left(Y\right)-1\right) + \mathbb{P}\left(\neg Y\right)\mathbb{E}\left[X|\neg Y\right] \\ &= -\mathbb{E}\left[X|Y\right]\mathbb{P}\left(\neg Y\right) + \mathbb{P}\left(\neg Y\right)\mathbb{E}\left[X|\neg Y\right] = \mathbb{P}\left(\neg Y\right)\left(\mathbb{E}\left[X|\neg Y\right] - \mathbb{E}\left[X|Y\right]\right) \\ \left|\mathbb{E}\left[X\right] - \mathbb{E}\left[X|Y\right]\right| &= \mathbb{P}\left(\neg Y\right)\underbrace{\left|\mathbb{E}\left[X|\neg Y\right] - \mathbb{E}\left[X|Y\right]\right|}_{\leq 1} \leq \mathbb{P}\left(\neg Y\right) .\end{aligned}$$

As a result

$$\left|\mathbb{E}_{S,A(S)}\left[\mathcal{L}_{\mathcal{D}}\left(A\left(S\right)\right)\right] - \mathbb{E}_{S,A(S)}\left[\mathcal{L}_{\mathcal{D}}\left(A\left(S\right)\right) \mid \text{consistent } S\right]\right| \leq \mathbb{P}_S(\text{inconsistent } S) .$$

Then, the required inequality is obtained by simply using the triangle inequality on

$$\left|\mathbb{E}_{S,A(S)}\left[\mathcal{L}_{\mathcal{D}}\left(A\left(S\right)\right)\right] - 2\varepsilon^\star\left(1-\varepsilon^\star\right)\right| .$$

■

Appendix E. Memorizing the label flips (proofs for Section 3)

In this section, we prove Theorem 46. We begin with an informal outline of the proof idea. Inspired by Manoj and Srebro's analysis [58], our proof of Theorem 46 is based on the concept of a *pseudorandom generator*, defined below.

Definition 24 (Pseudorandom generator) *Let $G\colon \{0,1\}^r \to \{0,1\}^R$ be a function, let $\mathcal{V}$ be a class of functions $V\colon \{0,1\}^R \to \{0,1\}$, let $\mathcal{D}$ be a distribution over $\{0,1\}^R$, and let $\epsilon > 0$. We say that G is an ϵ-*pseudorandom generator *(ϵ-PRG) for $\mathcal{V}$ with respect to $\mathcal{D}$ if for every $V \in \mathcal{V}$, we have*

$$\left|\mathbb{P}_{\mathbf{y}\sim\mathcal{D}}(V(\mathbf{y}) = 1) - \mathbb{P}_{\mathbf{u}\in\{0,1\}^r}(V(G(\mathbf{u})) = 1)\right| \le \epsilon,$$

where $\mathbf{u}$ is sampled uniformly at random from $\{0,1\}^r$.

To connect Definition 24 to Theorem 46, let $R = 2^{d_0}$. Let $\mathcal{V}$ be the class of all conjunctions of literals, such as $V(\mathbf{y}) = \mathbf{y}_1 \wedge \bar{\mathbf{y}}_2 \wedge \mathbf{y}_4$. Let $\hat{\mathcal{X}} = f^{-1}(\{0,1\})$. There is a function $V_f \in \mathcal{V}$ such that given the entire truth table of a NN $\tilde{h}$, the function V_f verifies that $\tilde{h}$ agrees with f on $\hat{\mathcal{X}}$. This function V_f is a conjunction of N_1 many variables and $(N - N_1)$ many negated variables.

Let $\alpha = N_1/N$, and let $\mathcal{D} = \mathrm{Ber}(\alpha)^R$. Suppose G is an ϵ-PRG for $\mathcal{V}$ with respect to $\mathcal{D}$, where $\epsilon < \mathbb{P}_{\mathbf{y}\sim\mathcal{D}}(V_f(\mathbf{y}) = 1)$. Then $\mathbb{P}_{\mathbf{u}\in\{0,1\}^r}(V_f(G(\mathbf{u})) = 1) \neq 0$, i.e., there exists some $\mathbf{u}^\star \in \{0,1\}^r$ such that $V_f(G(\mathbf{u}^\star)) = 1$. Therefore, if we let $\tilde{h}$ be a NN that computes the function

$$\tilde{h}(\mathbf{x}) = G(\mathbf{u}^\star)_{\mathbf{x}}, \tag{6}$$

then $\tilde{h}$ agrees with f on $\hat{\mathcal{X}}$. In the equation above, $G(\mathbf{u}^\star)_{\mathbf{x}}$ denotes the $\mathbf{x}$-th bit of $G(\mathbf{u}^\star)$, thinking of $\mathbf{x}$ as a number from 0 to $R - 1$ represented by its binary expansion.

There is a large body of well-established techniques for constructing PRGs. (See, for example, Hatami and Hoza's recent survey [35].) Therefore, constructing a suitable PRG might seem like a promising approach to proving Theorem 46. However, this approach is flawed. The issue concerns the seed length (r). According to the plan outlined above, the seed $\mathbf{u}^\star$ is effectively hard-coded into the neural network $\tilde{h}$, which means that, realistically, the number of weights in $\tilde{h}$ will be at least r. Meanwhile, for the plan above to make sense, our PRG's error parameter (ϵ) must satisfy

$$\epsilon < \mathbb{P}_{\mathbf{y}\sim\mathcal{D}}(V_f(\mathbf{y}) = 1) = 2^{-H(\alpha)\cdot N} \approx 2^{-\binom{N}{N_1}}. \tag{7}$$

Comparing (7) to Theorem 46, we see that we would need a PRG with seed length

$$r = (1 + o(1)) \cdot \log(1/\epsilon).$$

But this is too much to ask. There is no real reason to expect such a PRG to exist, even if we ignore explicitness considerations. Indeed, in some cases, it is provably impossible to achieve a seed length smaller than $(2 - o(1)) \cdot \log(1/\epsilon)$ [2].

To circumvent this issue, we will work with a more flexible variant of the PRG concept called a *hitting set generator* (HSG).

Definition 25 (Hitting set generator) *Let $G\colon \{0,1\}^r \to \{0,1\}^R$ be a function, let $\mathcal{V}$ be a class of functions $V\colon \{0,1\}^R \to \{0,1\}$, let $\mathcal{D}$ be a distribution over $\{0,1\}^R$, and let $\epsilon > 0$. We say that G is an ϵ-*hitting set generator *(ϵ-HSG) for $\mathcal{V}$ with respect to $\mathcal{D}$ if for every $V \in \mathcal{V}$ such that $\mathbb{P}_{\mathbf{y}\sim\mathcal{D}}(V(\mathbf{y}) = 1) > \epsilon$, there exists $\mathbf{u}^\star \in \{0,1\}^r$ such that $V(G(\mathbf{u}^\star)) = 1$.*

Definition 25 is weaker than Definition 24, but an HSG is sufficient for our purposes. Crucially, one can show nonconstructively that for every $\mathcal{V}$, $\mathcal{D}$, and ϵ, there exists an HSG with seed length

$$1 \cdot \log(1/\epsilon) + \log\log|\mathcal{V}| + O(1),$$

whereas the nonconstructive PRG seed length is $2 \cdot \log(1/\epsilon) + \cdots$. To prove Theorem 46, we will construct an *explicit* HSG for conjunctions of literals with respect to $\mathrm{Ber}(\alpha)^R$ with a seed length of $(1 + o(1)) \cdot \log(1/\epsilon) + \operatorname{polylog} R$. We will ensure that our HSG is "explicit enough" to enable computing the function $\tilde{h}$ defined by (6) using a constant-depth NN with approximately r many weights.

Our HSG construction uses established techniques from the pseudorandomness literature. In brief, we use k-wise independence to construct an initial HSG with a poor dependence on ϵ, and then we apply an error reduction technique due to Hoza and Zuckerman [38]. Details follow.

E.1. Preprocessing the input to reduce the dimension

Before applying an HSG as outlined above, the first step of the proof of Theorem 46 is actually a preprocessing step that reduces the dimension to approximately $2 \log N$. This step is not completely essential, but it helps to improve the dependence on d_0 in Theorem 46. The preprocessing step is based on a standard trick, namely, we treat the input as a vector in $\mathbb{F}_2^{d_0}$ and apply a random matrix, where $\mathbb{F}_2$ denotes the field with two elements. That is:

Definition 26 ($\mathbb{F}_2$-linear and $\mathbb{F}_2$-affine functions) *A function $C\colon \{0,1\}^d \to \{0,1\}^{d'}$ is $\mathbb{F}_2$-linear if it has the form*

$$C(\mathbf{x}) = \mathbf{W}\mathbf{x},$$

where $\mathbf{W} \in \{0,1\}^{d' \times d}$ and the arithmetic is mod 2. More generally, we say that C is $\mathbb{F}_2$-affine if it has the form

$$C(\mathbf{x}) = \mathbf{W}\mathbf{x} + \mathbf{b},$$

where $\mathbf{W} \in \{0,1\}^{d' \times d}$, $\mathbf{b} \in \{0,1\}^{d'}$, and the arithmetic is mod 2.

The following fact is standard; we include the proof only for completeness.

Lemma 27 (Preprocessing to reduce the dimension) *Let $d_0 \in \mathbb{N}$, let $\hat{\mathcal{X}} \subseteq \{0,1\}^{d_0}$, and let $N = |\hat{\mathcal{X}}|$. There exists an $\mathbb{F}_2$-linear function $C_0\colon \{0,1\}^{d_0} \to \{0,1\}^{2\lceil \log N \rceil}$ that is injective on $\hat{\mathcal{X}}$.*

Proof Pick $\mathbf{W} \in \{0,1\}^{2\lceil \log N \rceil \times d_0}$ uniformly at random and let $C_0(\mathbf{x}) = \mathbf{W}\mathbf{x}$. For each pair of distinct points $\mathbf{x}, \mathbf{x}' \in \hat{\mathcal{X}}$, we have

$$\mathbb{P}(\mathbf{W}\mathbf{x} = \mathbf{W}\mathbf{x}') = \mathbb{P}(\mathbf{W}(\mathbf{x} - \mathbf{x}') = \mathbf{0}) = 2^{-2\lceil \log N \rceil} < 1/N^2.$$

Therefore, by the union bound over all pairs $\mathbf{x}, \mathbf{x}'$, there is a nonzero chance that C_0 is injective on $\hat{\mathcal{X}}$. Therefore, there is some fixing of $\mathbf{W}$ such that C_0 is injective on $\hat{\mathcal{X}}$. ■

We will choose C_0 to be injective on the domain of f. That way, after applying C_0 to the input, our remaining task is to compute some other partial function $f'\colon \{0,1\}^{2\lceil \log N \rceil} \to \{0,1,\star\}$, namely, the function f' such that $f' \circ C_0 = f$. This function f' has the same domain size (N), and it takes the value 1 on the same number of points (N_1), so the net effect is that we have decreased the dimension

from d_0 down to $2\lceil \log N \rceil$. This same technique appears in the circuit complexity literature, along with more sophisticated variants. For example, see Jukna's textbook [45, Section 1.4.2].

To apply Lemma 27 in our setting, we rely on the well-known fact that $\mathbb{F}_2$-linear functions, and more generally $\mathbb{F}_2$-affine functions, can be computed by depth-two binary threshold networks. More precisely, we have the following.

Lemma 28 (Binary threshold networks computing $\mathbb{F}_2$-linear functions) *If $C\colon \{0,1\}^d \to \{0,1\}$ is the parity function or its negation, then there exists a depth-one binary threshold network $C_0\colon \{0,1\}^d \to \{0,1\}^{(d+2)}$ and a number $b \in \mathbb{R}$ such that for every $\mathbf{x} \in \{0,1\}^d$, we have*

$$C(x) = \mathbf{1}^T C_0(\mathbf{x}) + b,$$

where $\mathbf{1}$ denotes the all-ones vector. Moreover, every affine function $C\colon \{0,1\}^d \to \{0,1\}^{d'}$ can be computed by a depth-two binary threshold network with $d' \cdot (d+2)$ nodes in the hidden layer.

Proof First, suppose C is the parity function. For each $i \in [d]$, let $\phi_{\le i}\colon \{0,1\}^d \to \{0,1\}$ be the function

$$\phi_{\le i}(\mathbf{x}) = 1 \iff \sum_{j=1}^{d} \mathbf{x}_j \le i,$$

and similarly define $\phi_{\ge i}\colon \{0,1\}^d \to \{0,1\}$ by

$$\phi_{\ge i}(\mathbf{x}) = 1 \iff \sum_{j=1}^{d} \mathbf{x}_j \ge i.$$

Then

$$\phi_{\le 1}(\mathbf{x}) + \phi_{\ge 1}(\mathbf{x}) + \phi_{\le 3}(\mathbf{x}) + \phi_{\ge 3}(\mathbf{x}) + \cdots = \begin{cases} \lceil d/2 \rceil + 1 & \text{if } \mathsf{PARITY}(\mathbf{x}) = 1 \\ \lceil d/2 \rceil & \text{if } \mathsf{PARITY}(\mathbf{x}) = 0, \end{cases}$$

so we can take $b = -\lceil d/2 \rceil$. Now, suppose C is the negation of the parity function. This reduces to the case of the parity function because $1 - \mathsf{PARITY}(\mathbf{x}) = \mathsf{PARITY}(\mathbf{x}, 1)$. Finally, the "moreover" statement follows because if C is $\mathbb{F}_2$-affine, then every output bit of C is either the parity function or the negated parity function applied to some subset of the inputs. ■

Lemma 28 can be generalized to the case of any symmetric function instead of PARITY. This technique is well-known in the circuit complexity literature; for example, see the work of Hajnal, Maass, Pudlák, Szegedy, and Turán [32].

E.2. Threshold networks computing k-wise independent generators

One of the ingredients of our HSG will be a family of pairwise independent hash functions. We will use the following family, notable for its low computational complexity.

Lemma 29 (Affine pairwise independent hash functions) *For every $a, r \in \mathbb{N}$, there is a family $\mathcal{H}$ of hash functions $\mathrm{hash}\colon \{0,1\}^a \to \{0,1\}^r$ with the following properties.*

- $|\mathcal{H}| \le 2^{O(a+r)}$.

- *$\mathcal{H}$ is pairwise independent. That is, for every two distinct $\mathbf{w}, \mathbf{w}' \in \{0,1\}^a$, if we pick $\mathrm{hash} \in \mathcal{H}$ uniformly at random, then $\mathrm{hash}(\mathbf{w})$ and $\mathrm{hash}(\mathbf{w}')$ are independent and uniformly distributed over $\{0,1\}^r$.*
- *Each function $\mathrm{hash} \in \mathcal{H}$ is $\mathbb{F}_2$-affine.*

Proof See the work of Mansour, Nisan, and Tiwari [59, Claim 2.2]. ■

Remark 30 (Alternative hash families) *By Lemma 28, each function $\mathrm{hash} \in \mathcal{H}$ can be computed by a depth-two binary threshold network with $O(r^2a + ra^2)$ wires (weights). There exist alternative pairwise independent hash function families with lower wire complexity. In particular, one could use hash functions based on integer arithmetic [24], which can be implemented with wire complexity $(a+r)^{1+\gamma}$ for any arbitrarily small constant $\gamma > 0$ [72]. This would lead to slightly better width and wire complexity bounds in Theorem 46: each occurrence of $3/4$ could be replaced with $2/3 + \gamma$. However, the downside of this approach is that the depth of the network would increase to a very large constant depending on γ.*

Another ingredient of our HSG will be a threshold network computing a "k-wise uniform generator," defined below.

Definition 31 (k-wise uniform generator) *A k-wise uniform generator is a function $G\colon \{0,1\}^r \to \{0,1\}^R$ such that if we sample $\mathbf{u} \in \{0,1\}^r$ uniformly at random, then every k of the output coordinates of $G(\mathbf{u})$ are independent and uniform. In other words, G is a 0-PRG for $\mathcal{V}$ with respect to the uniform distribution, where $\mathcal{V}$ consists of all Boolean functions that only depend on k bits.*

Prior work has shown that k-wise uniform generators can be implemented by constant-depth threshold networks [36]. We will need to re-analyze the construction to get sufficiently fine-grained bounds. In the remainder of this subsection, we will prove the following.

Lemma 32 (Constant-depth k-wise uniform generator) *Let $k, R \in \mathbb{N}$ where R is a power of two. There exists a k-wise uniform generator $G\colon \{0,1\}^r \to \{0,1\}^R$, where $r = O(k \cdot \log R)$, such that for every $\mathbb{F}_2$-affine function $\mathrm{hash}\colon \{0,1\}^a \to \{0,1\}^r$, there exists a depth-5 binary threshold network $C\colon \{0,1\}^{a+\log R} \to \{0,1\}^{k \cdot \mathrm{polylog}\, R}$ with widths $\underline{d}$ satisfying the following.*

1. *For every $\mathbf{w} \in \{0,1\}^a$ and every $\mathbf{z} \in \{0,1\}^{\log R}$, we have*

$$G(\mathrm{hash}(\mathbf{w}))_{\mathbf{z}} = \mathsf{PARITY}(C(\mathbf{w}, \mathbf{z})),$$

   thinking of $\mathbf{z}$ as a number in $\{0, 1, \ldots, R-1\}$.

2. *The maximum width $\underline{d}_{\max}$ is at most $ak \cdot \mathrm{polylog}\, R$.*

3. *The total number of weights $w(\underline{d})$ is at most $(a+k) \cdot ak \cdot \mathrm{polylog}\, R$.*

Remark 33 (The role of the parity functions) *One can combine Lemma 32 with Lemma 28 to obtain threshold networks computing the function $(\mathbf{u}, \mathbf{z}) \mapsto G(\mathbf{u})_{\mathbf{z}}$. In Lemma 32, instead of describing a network that computes the function $(\mathbf{u}, \mathbf{z}) \mapsto G(\mathbf{u})_{\mathbf{z}}$, we describe a network C satisfying $G(\mathrm{hash}(\mathbf{w}))_{\mathbf{z}} = \mathsf{PARITY}(C(\mathbf{w}, \mathbf{z}))$. The only reason for this more complicated statement is that it leads to a slightly better depth complexity in Theorem 46.*

We reiterate that the proof of Lemma 32 heavily relies on prior work. For the most part, this prior work studies a Boolean circuit model that is closely related to, but distinct from, the "binary threshold network" model in which we are interested. We introduce the circuit model next.

Definition 34 ($\widehat{\mathrm{LT}}_L$ circuits) *An $\widehat{\mathrm{LT}}_L$ circuit is defined just like a depth-L binary threshold network (Definition 1), except that we allow arbitrary integer weights ($\mathbf{W}^{(l)} \in \mathbb{Z}^{d_l \times d_{l-1}}$); we allow arbitrary integer thresholds ($\mathbf{b}^{(l)} \in \mathbb{Z}^{d_l}$); and we do not allow any scaling ($\gamma^{(l)} = \mathbf{1}^{d_l}$). The* size *of the circuit is the sum of the absolute values of the weights, i.e.,*

$$\sum_{l=1}^{L} \sum_{i=1}^{d_l} \sum_{j=1}^{d_{l-1}} |\mathbf{W}_{ij}^{(l)}|.$$

Remark 35 (Parallel wires) *In the context of circuit complexity, it is perhaps more natural to stipulate that the weights are always $\{\pm 1\}$; there can be any number of* parallel wires *between two nodes, including zero; and the size of the circuit is the total number of wires. This is completely equivalent to Definition 34.*

The proof of Lemma 32 relies on circuits performing arithmetic. A long line of research investigated the depth complexity of (iterated) integer multiplication [9, 16, 32, 37, 69, 72, 77–79], culminating in the following result by Siu and Roychowdhury [78].

Theorem 36 (Iterated multiplication in depth four [78]) *For every $n \in \mathbb{N}$, there exists an $\widehat{\mathrm{LT}}_4$ circuit of size $\mathrm{poly}(n)$ that computes the product of n given n-bit integers.*

By a standard trick [26], Theorem 36 implies circuits of the same complexity that compute the iterated product of *polynomials* over $\mathbb{F}_2$. We include a proof sketch for completeness.

Corollary 37 (Iterated multiplication of polynomials over $\mathbb{F}_2$) *For every $n \in \mathbb{N}$, there exists an $\widehat{\mathrm{LT}}_4$ circuit of size $\mathrm{poly}(n)$ that computes the product of n given polynomials in $\mathbb{F}_2[x]$, each of which has degree less than n and is represented by an n-bit vector of coefficients.*

Proof [Proof sketch] Think of the given polynomials as polynomials over $\mathbb{Z}$, say $q_1(x), \ldots, q_n(x)$. If we evaluate one of these polynomials on a power of two, say $q_i(2^s)$, and then write the output in binary, the resulting string consists of the coefficients of q_i, with $s-1$ zeroes inserted between every two bits. Therefore, by using the $\mathrm{poly}(ns)$-size circuit of Therem 36 (with some of its inputs fixed to zeroes), we can compute the product $q_1(2^s) \cdot q_2(2^s) \cdots q_n(2^s) = q(2^s)$, where $q = q_1 \cdot q_2 \cdots q_n$. Every coefficient of q is a nonnegative integer bounded by n^n, so if we choose $s = \lceil n \log n \rceil$, then the binary expansion of $q(2^s)$ is the concatenation of all of the binary expansions of the coefficients of q. To reduce mod 2, we simply discard all but the lowest-order bit of each of those coefficients. ■

At this point, we are ready to construct a circuit that computes a k-wise uniform generator. The construction is based on the work of Healy and Viola [36].

Lemma 38 (A k-wise uniform generator in the $\widehat{\mathrm{LT}}_L$ model) *Let $k, R \in \mathbb{N}$ where R is a power of two. There exists a k-wise uniform generator $G\colon \{0,1\}^r \to \{0,1\}^R$, an $\mathbb{F}_2$-linear function $C_0\colon \{0,1\}^{\log R} \to \{0,1\}^{O(\log R \cdot \log k)}$, and an $\widehat{\mathrm{LT}}_4$ circuit $C_1\colon \{0,1\}^{O(k\cdot\log R)} \to \{0,1\}^{O(k\cdot\log R)}$ with the following properties.*

- *The seed length is $r = O(k \cdot \log R)$.*
- *For every seed $\mathbf{u} \in \{0,1\}^r$ and every $\mathbf{z} \in \{0,1\}^{\log R}$, we have*

$$G(\mathbf{u})_{\mathbf{z}} = \mathsf{PARITY}(C_1(\mathbf{u}, C_0(\mathbf{z}))),$$

  thinking of $\mathbf{z}$ as a number in $\{0, 1, \ldots, R-1\}$.
- *The circuit C_1 has size $k \cdot \operatorname{polylog} R$.*

Proof If $k \geq R$, the lemma is trivial, so assume $k < R$. We use the following standard example of a k-wise independent generator [1, 22]. Let $n = \log R$, let $E(x) \in \mathbb{F}_2[x]$ be an irreducible polynomial of degree n, and let $\mathbb{F}_{2^n}$ be the finite field consisting of all polynomials in $\mathbb{F}_2[x]$ modulo $E(x)$. The seed of the generator is interpreted as a list of field elements: $\mathbf{u} = (p_0, p_1, \ldots, p_{k-1}) \in \mathbb{F}_{2^n}^k$. Each index $\mathbf{z} \in \{0, 1, \ldots, R-1\}$ can be interpreted as a field element $\mathbf{z} \in \mathbb{F}_{2^n}$. The output of the generator is given by

$$G(\mathbf{u})_{\mathbf{z}} = \text{the lowest order bit of } p_0 \cdot \mathbf{z}^0 + p_1 \cdot \mathbf{z}^1 + \cdots + p_{k-1} \cdot \mathbf{z}^{k-1},$$

where the arithmetic takes place in $\mathbb{F}_{2^n}$.

To study the circuit complexity of this generator, let us first focus on a single term $p_i \cdot \mathbf{z}^i$. Since we are thinking of $\mathbf{z}$ as a field element $\mathbf{z} \in \mathbb{F}_{2^n}$, we can also think of it as a polynomial $\mathbf{z}(x) \in \mathbb{F}_2[x]$ of degree less than n. Write $\mathbf{z}(x) = \sum_{j=0}^{n-1} \mathbf{z}_j \cdot x^j$. We compute the power $\mathbf{z}^i$ by a "repeated squaring" approach. Write $i = \sum_{m \in M} 2^m$, where $M \subseteq \{0, 1, \ldots, \lfloor \log i \rfloor\}$. Then

$$p_i(x) \cdot \mathbf{z}(x)^i = p_i(x) \cdot \prod_{m \in M} \left(\sum_{j=0}^{n-1} \mathbf{z}_j \cdot x^j \right)^{2^m} = p_i(x) \cdot \prod_{m \in M} \sum_{j=0}^{n-1} \mathbf{z}_j \cdot x^{j \cdot 2^i},$$

since we are working in characteristic two. For each $m \in M$ and each $j < n$, let $e_{m,j}(x) = x^{j \cdot 2^m} \bmod E(x)$, a polynomial of degree less than n. That way,

$$p_i(x) \cdot \mathbf{z}(x)^i \equiv p_i(x) \cdot \prod_{m \in M} \sum_{j=0}^{n-1} \mathbf{z}_j \cdot e_{m,j}(x) \pmod{E(x)}. \tag{8}$$

The function $C_0(\mathbf{z})$ computes $\sum_{j=0}^{n-1} \mathbf{z}_j \cdot e_{m,j}$ for every $m \in \{0, 1, \ldots, \lfloor \log k \rfloor\}$. This function is $\mathbb{F}_2$-linear, because for each $m \in \{0, 1, \ldots, \lfloor \log k \rfloor\}$ and each $s < n$, the s-th bit of $\sum_{j=0}^{n-1} \mathbf{z}_j \cdot e_{m,j}$ is given by

$$\bigoplus_{j : e_{m,j,s} = 1} \mathbf{z}_j,$$

where $e_{m,j,s}$ denotes the s-th coefficient of the polynomial $e_{m,j}$.

Next, the circuit C_1 applies k copies of the iterated multiplication circuit from Corollary 37, in parallel, to compute the polynomial on the right-hand side of (8) for each $0 \le i < k$. Each iterated multiplication circuit has size $\operatorname{polylog} R$, so altogether, C_1 has size $k \cdot \operatorname{polylog} R$.

At this point, we have computed polynomials $r_0, r_1, \ldots, r_{k-1} \in \mathbb{F}_2[x]$, each of degree $O(n \log k)$, such that $r_i(x) \equiv p_i(x) \cdot \mathbf{z}(x)^i \pmod{E(x)}$. Next, we need to sum these terms up, reduce mod $E(x)$, and output the lowest-order bit. For each $j \le O(n \log k)$, let r_{ij} be the x^j coefficient of r_i. Our circuit needs to output the lowest-order bit of

$$\sum_{i=0}^{k-1} r_i \bmod E(x) = \sum_{i=0}^{k-1} \sum_{j=0}^{O(n \log k)} r_{ij} \cdot e_{0,j}.$$

Now, we are working over characteristic two, so $\sum$ means bitwise XOR. In other words, the output is given by

$$\bigoplus_{j : e_{0,j,0}=1} \bigoplus_{i=0}^{k-1} r_{ij},$$

i.e., it is the parity function applied to some subset of the output bits of C_1. To complete the proof, modify C_1 by deleting the unused output gates. ■

We have almost completed the proof of Lemma 32. The last step is to bridge the gap between $\widehat{\mathrm{LT}}_L$ circuits and binary threshold networks. We do so via the following lemma.

Lemma 39 (Simulating $\widehat{\mathrm{LT}}_L$ circuits using binary threshold networks) *Let $L \ge 1$ be a constant. Let $C_0 \colon \{0,1\}^{d_0} \to \{0,1\}^{d_1}$ be an $\mathbb{F}_2$-affine function, and let $C_1 \colon \{0,1\}^{d_1} \to \{0,1\}^{d_2}$ be an $\widehat{\mathrm{LT}}_L$ circuit of size S. Then the composition $C_1 \circ C_0$ can be computed by a depth-$(L+1)$ binary threshold network with widths $\underline{d}$ satisfying the following.*

1. *The maximum width $\underline{d}_{\max}$ is at most $S \cdot (d_0 + 2)$.*
2. *The total number of weights $w(\underline{d})$ is at most $O(S^2 d_0 + S d_0^2)$.*

Proof Let us define the *cost* of a layer in an $\widehat{\mathrm{LT}}_L$ circuit to be the sum of the absolute values of the weights in that layer, so the size of the circuit is the sum of the costs. Lemma 28 implies that $C_1 \circ C_0 = C_1' \circ C_0'$, where C_0' is a depth-one binary threshold network and C_1' is an $\widehat{\mathrm{LT}}_L$ circuit in which the first layer has cost at most $S \cdot (d_0 + 2)$ and all subsequent layers have cost at most S.

To complete the proof, let us show by induction on L that in general, if C_0' is a depth-one binary threshold network and C_1' is an $\widehat{\mathrm{LT}}_L$ circuit in which the layers have costs $S_1, S_2, \ldots, S_L$, then $C_1' \circ C_0'$ can be computed by a depth-$(L+1)$ binary threshold network in which the layers after the input layer have widths $S_1, S_2, \ldots, S_L$. Let us write C_1' as $C_3 \circ C_2$, where C_3 is the last layer of C_1'. By induction, $C_2 \circ C_0'$ can be computed by a depth-L binary threshold network C in which the layers after the input layer have widths $S_1, S_2, \ldots, S_{L-1}$. Now let us modify C_3 and C so that every wire in C_3 has weight either 0 or 1. If a wire in C_3 has an integer weight $w \notin \{0,1\}$, then we make $|w|$ many copies of the appropriate output gate of C, negate them if $w < 0$, and then split the wire into $|w|$ many wires, each with weight $+1$. This process has no effect on the cost of C_3. The process could potentially increase the width of the output layer of C, but its width will not exceed S_L,

the cost of C_3. After this modification, we can simply think of C_3 as one more layer in our binary threshold network. ■

Lemma 32 follows immediately from Lemmas 38 and 39.

E.3. A hitting set generator with a non-optimal dependence on ϵ

In this subsection, we will use the k-wise independent generators that we developed in the previous subsection to construct our first HSG:

Lemma 40 (Non-optimal HSG for conjunctions of literals) *Let R be a power of two and let $\alpha, \epsilon \in (0,1)$. Assume that $1/R \leq \alpha \leq 1-1/R$. Let $\mathcal{V}$ be the class of functions $V\colon \{0,1\}^R \to \{0,1\}$ that can be expressed as a conjunction of literals. There exists a generator $G\colon \{0,1\}^r \to \{0,1\}^R$ satisfying the following.*

1. *For every $V \in \mathcal{V}$, if $\mathbb{P}_{\mathbf{y}\sim\mathrm{Ber}(\alpha)^R}(V(y)=1) \geq 2\epsilon$, then $\mathbb{P}_{\mathbf{u}\in\{0,1\}^r}(V(G(\mathbf{u}))=1) \geq \epsilon$.*
2. *The seed length is $r = O(\log(1/\epsilon)\cdot\log^2 R)$.*
3. *For every $\mathbb{F}_2$-affine function $\mathrm{hash}\colon \{0,1\}^a \to \{0,1\}^r$, the function $C(\mathbf{w},\mathbf{z}) = G(\mathrm{hash}(\mathbf{w}))_{\mathbf{z}}$ can be computed by a depth-8 binary threshold network with widths $\underline{d}$ such that the maximum width $\underline{d}_{\max}$ at most $a\cdot\log(1/\epsilon)\cdot\mathrm{polylog}\,R$ and the total number of weights $w(\underline{d})$ is at most $(\log(1/\epsilon)\cdot a^2 + \log^2(1/\epsilon)\cdot a)\cdot\mathrm{polylog}\,R$.*

Remark 41 *The parameters of Lemma 40 are not yet sufficient to prove Theorem 46. Remember, we need the number of weights to be only $(1+o(1))\cdot\log(1/\epsilon)$. On the other hand, Item 1 is stronger than what the HSG definition requires. This will enable us to improve the seed length of the generator later.*

The proof of Lemma 40 is based on the work of Even, Goldreich, Luby, Nisan, and Veličković [27]. In particular, we use the following lemma from their work.

Lemma 42 (Implications of k-wise independence [27]) *Let $X_1,\ldots,X_R$ be independent $\{0,1\}$-valued random variables. Let $\tilde{X}_1,\ldots,\tilde{X}_R$ be k-wise independent $\{0,1\}$-valued random variables such that $\tilde{X}_i$ is distributed identically to X_i for every i. Then*

$$|\mathbb{P}(X_1 = X_2 = \cdots = X_R = 1) - \mathbb{P}(\tilde{X}_1 = \tilde{X}_2 = \cdots = \tilde{X}_R = 1)| \leq 2^{-\Omega(k)}.$$

Proof [Proof of Lemma 40] Let Q be the smallest positive integer such that $Q \geq 4R^2$ and $\log\log Q$ is an integer. Let $\phi\colon \{0,1,\ldots,Q-1\} \to \{0,1\}$ be the function

$$\phi(x) = 1 \iff x \leq \alpha\cdot Q.$$

We think of ϕ as a function $\phi\colon \{0,1\}^{\log Q} \to \{0,1\}$ by representing x in binary.

Let $\boldsymbol{\phi}\colon \{0,1\}^{R\log Q} \to \{0,1\}^R$ be R copies of ϕ applied to R disjoint input blocks. Let $G_0\colon \{0,1\}^r \to \{0,1\}^{R\log Q}$ be a k-wise independent generator for a suitable value $k = O(\log(1/\epsilon)\cdot\log R)$. Our generator G is the composition $\boldsymbol{\phi}\circ G_0$.

Now let us prove that G has the claimed properties. The seed length bound is clear. Now let us analyze the computational complexity of G. To compute $G(\mathrm{hash}(\mathbf{w}))_{\mathbf{z}}$, we begin by computing $C_1(\mathbf{w}, \mathbf{z}\log Q + i)$ for every $i \in \{0,1,\ldots,\log Q - 1\}$, all in parallel, where C_1 is the depth-5 network from Lemma 32. Since $\log Q$ is a power of two, the binary expansions of the numbers $\mathbf{z}\log Q, \mathbf{z}\log Q + 1, \mathbf{z}\log Q + 2, \ldots, \mathbf{z}\log Q + \log Q - 1$ simply consist of $\mathbf{z}$ followed by all possible bitstrings of length $\log\log Q$. The maximum width of one of these layers is bounded by $ak\cdot\mathrm{polylog}\,R = a\cdot\log(1/\epsilon)\cdot\mathrm{polylog}\,R$, and the total number of weights among these layers is at

most $(a+k) \cdot ak \cdot \operatorname{polylog} R = (a + \log(1/\epsilon)) \cdot a \cdot \log(1/\epsilon) \cdot \operatorname{polylog} R$. Furthermore, the number of output bits is $\log(1/\epsilon) \cdot \operatorname{polylog} R$.

Next, recall that to compute a single bit of the output of G_0, we need to apply the parity function to the outputs of C_1. Therefore, to compute an output bit of our generator G, we need to apply an $\mathbb{F}_2$-linear function followed by ϕ. Observe that ϕ can be computed by a depth-two "AC^0 circuit," i.e., a circuit consisting of unbounded-fan-in AND and OR gates applied to literals, in which the total number of gates is $O(\log Q) = O(\log R)$. This can be viewed as a special case of an $\widehat{\mathrm{LT}}_2$ circuit of size $O(\log^2 R)$. Therefore, by Lemma 39, the $\mathbb{F}_2$-linear function followed by ϕ can be computed by a depth-3 binary threshold network in which every layer has width at most $\log(1/\epsilon) \cdot \operatorname{polylog} R$ and the total number of weights is at most $\log^2(1/\epsilon) \cdot \operatorname{polylog} R$. This completes the analysis of the computational complexity of G.

Next, let us prove the correctness of G, i.e., let us prove Item 1 of Lemma 40. Let $V \in \mathcal{V}$ and assume $\mathbb{P}_{\mathbf{y} \sim \mathrm{Ber}(\alpha)^R}(V(\mathbf{y}) = 1) \geq 2\epsilon$. Since V is a conjunction of literals, we can write $V(\mathbf{y}) = V_1(\mathbf{y}_1) \cdot V_2(\mathbf{y}_2) \cdots V_R(\mathbf{y}_R)$ for some functions $V_1, V_2, \ldots, V_R \colon \{0,1\} \to \{0,1\}$.

We will analyze $\mathbb{P}_{\mathbf{u} \in \{0,1\}^r}(V(G(\mathbf{u})) = 1)$ in two stages. First, we compare $V(\boldsymbol{\phi}(G_0(\mathbf{u})))$ to $V(\boldsymbol{\phi}(\bar{\mathbf{y}}))$, where $\bar{\mathbf{y}} \in \{0,1\}^{R \log Q}$ is uniform random. Then, in the second stage, we will compare $V(\boldsymbol{\phi}(\bar{\mathbf{y}}))$ to $V(\mathbf{y})$, where $\mathbf{y} \sim \mathrm{Ber}(\alpha)^R$.

For the first stage, we are in the situation of Lemma 42, because the R many $(\log Q)$-bit blocks of $G_0(\mathbf{u})$ are $(k/\log Q)$-wise independent. Therefore,

$$\left| \mathbb{P}_{\mathbf{u} \in \{0,1\}^r}(V(G(\mathbf{u})) = 1) - \mathbb{P}_{\bar{\mathbf{y}} \in \{0,1\}^{R \log Q}}(V(\boldsymbol{\phi}(\bar{\mathbf{y}})) = 1) \right| \leq \exp\left(-\Omega\left(\frac{k}{\log Q}\right)\right) \leq 0.5\epsilon,$$

provided we choose a suitable value $k = O(\log(1/\epsilon) \cdot \log R)$.

Now, for the second stage, observe that if we sample $\bar{\mathbf{y}} \in \{0,1\}^{\log Q}$ uniformly at random, then $|\mathbb{P}(\phi(\bar{\mathbf{y}}) = 1) - \alpha| \leq \frac{1}{Q} \leq \frac{1}{4R^2}$. For each i, since $1/R \leq \alpha \leq 1 - 1/R$, we have

$$\begin{aligned} \mathbb{P}_{\bar{\mathbf{y}} \in \{0,1\}^{\log Q}}(V_i(\phi(\bar{\mathbf{y}})) = 1) &\geq \mathbb{P}_{\mathbf{y} \sim \mathrm{Ber}(\alpha)}(V_i(\mathbf{y}) = 1) - \frac{1}{4R^2} \\ &\geq \left(1 - \frac{1}{4R}\right) \cdot \mathbb{P}_{\mathbf{y} \sim \mathrm{Ber}(\alpha)}(V_i(\mathbf{y}) = 1). \end{aligned}$$

Therefore,

$$\begin{aligned} \mathbb{P}_{\bar{\mathbf{y}} \in \{0,1\}^{R \log Q}}(V(\boldsymbol{\phi}(\bar{\mathbf{y}})) = 1) &\geq \left(1 - \frac{1}{4R}\right)^R \cdot \mathbb{P}_{\mathbf{y} \sim \mathrm{Ber}(\alpha)^R}(V(\mathbf{y}) = 1) \\ &\geq 1.5\epsilon \end{aligned}$$

by Bernoulli's inequality. Combining the bounds from the two stages completes the proof. ■

E.4. Networks for computing functions that are constant on certain intervals

At this point, we have constructed an HSG for conjunctions of literals with a non-optimal dependence on the threshold parameter ϵ (Lemma 40). To improve the dependence on ϵ, we will use a technique introduced by Hoza and Zuckerman [38]. They introduced this "error-reduction" technique in the context of derandomizing general space-bounded algorithms, but it is simpler in our context (conjunctions of literals).

The basic idea is as follows. Let V be a conjunction of literals with a low acceptance probability: $\mathbb{P}_{\mathbf{y}\sim\mathrm{Ber}(\alpha)^R}(V(\mathbf{y}) = 1) = \epsilon$. We will split V up as a product,

$$V(\mathbf{y}) = V^{(0)}(\mathbf{y}^{(0)}) \cdot V^{(1)}(\mathbf{y}^{(1)}) \cdots V^{(T-1)}(\mathbf{y}^{(T-1)}),$$

where each $V^{(i)}$ is a conjunction of literals with a considerably higher acceptance probability:

$$\mathbb{P}_{\mathbf{y}^{(i)}\sim\mathrm{Ber}(\alpha)^{R_i}}(V^{(i)}(\mathbf{y}^{(i)}) = 1) \approx \epsilon_0 \gg \epsilon.$$

We choose $V^{(0)}$ to be the conjunction of the first few literals in V; $V^{(1)}$ is the conjunction of the next few literals; etc. To hit a single $V^{(i)}$, we can use our initial HSG with a relatively high threshold parameter (ϵ_0). Then, we use pairwise independent hash functions to "recycle" the seed of our initial HSG from one $V^{(i)}$ to the next.

To implement this technique, one of the ingredients we need is a network that figures out which block $V^{(i)}$ contains a particular given index $\mathbf{z} \in \{0, 1, \ldots, R-1\}$. In this subsection, we describe networks that handle that key ingredient. The constructions are elementary and straightforward.

First, we review standard circuits for integer comparisons.

Lemma 43 (DNFs for comparing integers) *Let R be a power of two, let $I \subseteq [0, R)$ be an interval, and let $g_I \colon \{0,1\}^{\log R} \to \{0,1\}$ be the indicator function for $I \cap \{0, 1, \ldots, R-1\}$ (identifying numbers with their binary expansions). Then g_I can be expressed as a DNF formula consisting of $O(\log^2 R)$ terms.*

Proof First, consider the case that $I = [0, r)$ for some $r \in \{1, 2, \ldots, R\}$. If $r = R$, then the lemma is trivial, so assume $r < R$. Let S be the set of indices at which the binary expansion of r has a one. For each $i \in S$, we introduce a term that asserts that the input $\mathbf{z}$ agrees with the binary expansion of r prior to position i, and then $\mathbf{z}$ has a zero in position i. The disjunction of these $|S|$ many terms computes g_I.

The case $I = [\ell, R)$ for some $\ell \in \{0, 1, \ldots, R-1\}$ is symmetric. Finally, in the general case, we can assume that I is an intersection of an interval of the form $[\ell, R)$ with an interval of the form $[0, r)$. Therefore, g_I can be expressed in the form $\mathsf{AND}_2 \circ \mathsf{OR}_{\log R} \circ \mathsf{AND}_{\log R}$, where AND_k / OR_k denotes an AND / OR gate with fan-in k. To complete the proof, observe that every $\mathsf{AND}_a \circ \mathsf{OR}_b$ formula can be re-expressed as an $\mathsf{OR}_{b^a} \circ \mathsf{AND}_a$ formula. ■

Lemma 44 (Computing a function that is constant on intervals) *Let T and R be powers of two. Suppose the interval $[0, R)$ has been partitioned into T subintervals, say $[0, R) = I_0 \cup I_1 \cup \cdots \cup I_{T-1}$. Let $g \colon \{0, 1, \ldots, R-1\} \to \{0,1\}^a$ be a function that is constant on each subinterval I_j. Then for every $\mathbb{F}_2$-affine function $C_0 \colon \{0,1\}^{d_0} \to \{0,1\}^{\log R}$, there is a depth-6 binary threshold network $C \colon \{0,1\}^{d_0} \to \{0,1\}^{a+\log R}$ with widths $\underline{d}$ satisfying the following.*

1. *For every $\mathbf{x} \in \{0,1\}^{d_0}$, we have*

$$C(\mathbf{x}) = (g(C_0(\mathbf{x})), C_0(\mathbf{x})).$$

2. *The maximum width $\underline{d}_{\max}$ is at most $O(T \cdot \log^3 R + a + d_0 \cdot \log R)$.*

3. *The total number of weights* $w(\underline{d})$ *is at most*

$$aT + O(T \cdot \log^4 R + d_0^2 \cdot \log R + d_0 \cdot \log^2 R + a \cdot \log R).$$

We emphasize that the leading term in the weights bound is aT, with a coefficient of 1. This is crucial. It is also important that the weights bound has only a linear dependence on T, the number of intervals.

Proof We begin by computing $C_0(\mathbf{x})$ and the negations of all of those bits. By Lemma 28, we can compute these bits using a depth-two network where the hidden layer has width $O(d_0 \cdot \log R)$ and the output layer has width $O(\log R)$.

Let $\mathbf{z} = C_0(\mathbf{x}) \in \{0,1\}^{\log R}$, and think of $\mathbf{z}$ as a number $\mathbf{z} \in \{0, 1, \ldots, R-1\}$. Our next goal is to compute the $(\log T)$-bit binary expansion of the unique $j_* \in \{0, 1, \ldots, T-1\}$ such that $\mathbf{z} \in I_{j_*}$. To do so, for each position $i \in \{0, 1, \ldots, \log T - 1\}$, let S_i be the set of $j \in \{0, 1, \ldots, T-1\}$ such that j has a 1 in position i of its binary expansion. We have a disjunction, over all $j \in S_i$, of the DNF computing g_{I_j} from Lemma 43. We also compute all the negations of the bits of j_*, and we also copy $\mathbf{z}$. Altogether, this is a depth-two network where the hidden layer has width $O(T \cdot \log T \cdot \log^2 R) = O(T \cdot \log^3 R)$ and the output layer has width $O(\log R)$.

Our final goal is to compute $g(\mathbf{z})$, which can be written in the form $g'(j_*)$ since g is constant on each subinterval. We use a "brute-force DNF" to compute g'. First, for every $j \in \{0, 1, \ldots, T-1\}$, we have an AND gate that checks whether $j_* = j$. Then each output bit of g' is a disjunction of some of those AND gates. We also copy $\mathbf{z}$. Altogether, this is a depth-two network where the hidden layer has width $T + \log R$ and the output layer has width $a + \log R$. ■

E.5. Error reduction

In this subsection, we improve our HSG's dependence on ϵ, as described in the previous subsection. The following theorem should be compared to Lemma 40. As discussed previously, the proof is based on a technique due to Hoza and Zuckerman [38].

Theorem 45 (HSG with near-optimal dependence on ϵ) *Let R be a power of two and let $\alpha, \epsilon \in (0,1)$. Assume that $1/R \leq \alpha \leq 1 - 1/R$. Let $\mathcal{V}$ be the class of functions $V\colon \{0,1\}^R$ that can be expressed as a conjunction of literals. There exists a generator $G\colon \{0,1\}^r \to \{0,1\}^R$ satisfying the following.*

1. *G is an ϵ-HSG for $\mathcal{V}$ with respect to $\mathrm{Ber}(\alpha)^R$. That is, if $\mathbb{P}_{\mathbf{y}\sim\mathrm{Ber}(\alpha)^R}(V(\mathbf{y}) = 1) > \epsilon$ for every $V \in \mathcal{V}$, then there exists a seed $\sigma \in \{0,1\}^r$ such that $V(G(\sigma)) = 1$.*
2. *The seed length is $r = \log(1/\epsilon) + \log^{3/4}(1/\epsilon) \cdot \operatorname{polylog} R$.*
3. *For every $\mathbb{F}_2$-affine function $C_0\colon \{0,1\}^{d_0} \to \{0,1\}^{\log R}$ and every fixed seed $\sigma \in \{0,1\}^r$, the function $\tilde{h}(\mathbf{x}) = G(\sigma)_{C_0(\mathbf{x})}$ can be computed by a depth-14 binary threshold network with widths $\underline{d}$ such that the maximum width $\underline{d}_{\max}$ is at most*

$$\log^{3/4}(1/\epsilon) \cdot \operatorname{polylog} R + O(d_0 \cdot \log R),$$

and the total number of weights $w(\underline{d})$ is at most

$$\log(1/\epsilon) + \log^{3/4}(1/\epsilon) \cdot \operatorname{polylog} R + O(d_0^2 \cdot \log R + d_0 \cdot \log^2 R).$$

Proof First we will describe the construction of G; then we will verify its seed length and computational complexity; and finally we will verify its correctness.

Construction. Let T be the smallest power of two such that $T \geq \log^{3/4}(1/\epsilon)$. Let

$$\epsilon_0 = \frac{\epsilon^{1/T}}{2R},$$

and note that $\log(1/\epsilon_0) = \Theta(\log^{1/4}(1/\epsilon) + \log R)$. Let $G_0\colon \{0,1\}^{r_0} \to \{0,1\}^R$ be the generator of Lemma 40 with error parameter ϵ_0, i.e., for every $V \in \mathcal{V}$, if $\mathbb{P}_{\mathbf{y}\sim\mathrm{Ber}(\alpha)^R}(V(\mathbf{y}) = 1) \geq 2\epsilon_0$, then $\mathbb{P}_{\mathbf{u}\in\{0,1\}^{r_0}}(V(G_0(\mathbf{u})) = 1) \geq \epsilon_0$. Let a be the smallest positive integer such that $2^a > R/\epsilon_0$. Let $\mathcal{H}$ be the family of $\mathbb{F}_2$-affine hash functions $\mathrm{hash}\colon \{0,1\}^a \to \{0,1\}^{r_0}$ from Lemma 29.

A seed for our generator G consists of a function $\mathrm{hash} \in \mathcal{H}$, inputs $\mathbf{w}^0, \ldots, \mathbf{w}^{T-1} \in \{0,1\}^a$, and nonnegative integers $0 = \ell_0 \leq \ell_1 \leq \cdots \leq \ell_T = R$. Given this data $\sigma = (\mathrm{hash}, \mathbf{w}^0, \ldots, \mathbf{w}^{T-1}, \ell_0, \ldots, \ell_T)$, the output of the generator is given by

$$G(\sigma) = G_0(\mathrm{hash}(\mathbf{w}^0))_{0\cdots\ell_1-1} \,\&\, G_0(\mathrm{hash}(\mathbf{w}^1))_{\ell_1\cdots\ell_2-1} \,\&\, \cdots \,\&\, G_0(\mathrm{hash}(\mathbf{w}^{T-1}))_{\ell_{T-1}\cdots\ell_T-1}.$$

In the equation above, $\mathbf{y}_{a\cdots b}$ denotes the substring of $\mathbf{y}$ consisting of the bits at positions $a, a+1, a+2, \ldots, b$, and $\&$ denotes concatenation.

Seed length and computational complexity. Since $|\mathcal{H}| \leq 2^{O(a+r_0)}$, the description length of hash is $O(a + r_0)$. The description length of $\mathbf{w}^0, \ldots, \mathbf{w}^{T-1}$ is aT, and the description length of $\ell^0, \ldots, \ell^T$ is $O(T \log R)$. By our choices of a and ϵ_0, we have

$$a \leq \log(1/\epsilon_0) + O(\log R) = \frac{\log(1/\epsilon)}{T} + O(\log R).$$

Furthermore, by Lemma 40, we have

$$r_0 = O(\log(1/\epsilon_0) \cdot \log^2 R).$$

Therefore, the overall seed length of our generator is

$$aT + O(r_0 + T \log R + a) \leq \log(1/\epsilon) + \log^{3/4}(1/\epsilon) \cdot \operatorname{polylog} R.$$

To analyze the computational complexity, fix an arbitrary seed

$$\sigma = (\text{hash}, \mathbf{w}^0, \ldots, \mathbf{w}^{T-1}, \ell_0, \ldots, \ell_T).$$

The numbers $\ell_0, \ldots, \ell_T$ partition the interval $[0, R)$ into subintervals, namely $[0, R) = [\ell_0, \ell_1) \cup [\ell_1, \ell_2) \cup \cdots \cup [\ell_{T-1}, \ell_T)$. Define $g \colon \{0, 1, \ldots, R-1\} \to \{0,1\}^a$ by the rule

$$g(\mathbf{z}) = \mathbf{w}^j \text{ where } j \text{ is such that } \mathbf{z} \in [\ell_j, \ell_{j+1}).$$

Then g is constant on each subinterval $[\ell_j, \ell_{j+1})$, so we may apply Lemma 44 to obtain a depth-6 binary threshold network $C_1 \colon \{0,1\}^{d_0} \to \{0,1\}^{a+\log R}$ computing the function $C(\mathbf{x}) = (g(C_0(\mathbf{x})), C_0(\mathbf{x}))$. In this network, every layer has width at most

$$O(T \cdot \log^3 R + a + d_0 \cdot \log R) = \log^{3/4}(1/\epsilon) \cdot \operatorname{polylog} R + O(d_0 \cdot \log R),$$

and the total number of weights is at most

$$\begin{aligned} &aT + O(T \cdot \log^4 R + d_0^2 \cdot \log R + d_0 \cdot \log^2 R + a \cdot \log R) \\ &\quad \leq \log(1/\epsilon) + \log^{3/4}(1/\epsilon) \cdot \operatorname{polylog} R + O(d_0^2 \cdot \log R + d_0 \cdot \log^2 R). \end{aligned}$$

Let $\mathbf{z} = C_0(\mathbf{x})$, and let $\mathbf{w} = g(\mathbf{z})$. Our remaining goal is compute $G(\sigma)_{\mathbf{z}}$, which is equal to $G_0(\text{hash}(\mathbf{w}))_{\mathbf{z}}$. To do so, we use the network guaranteed to exist by Lemma 40. This network, which we call C_2, has depth 8. Every layer in this network has width at most

$$a \cdot \log(1/\epsilon_0) \cdot \operatorname{polylog} R = \sqrt{\log(1/\epsilon)} \cdot \operatorname{polylog} R.$$

The total number of weights in this network is at most

$$(\log(1/\epsilon_0) \cdot a^2 + \log^2(1/\epsilon_0) \cdot a) \cdot \operatorname{polylog} R = \log^{3/4}(1/\epsilon) \cdot \operatorname{polylog} R.$$

Composing C_2 with C_1 completes the analysis of the computational complexity of our HSG.

Correctness. Finally, let us prove the correctness of our HSG. For convenience, for any $n \in \mathbb{N}$ and any function $V\colon \{0,1\}^n \to \{0,1\}$, we write $\mathbb{E}(V)$ to denote the quantity $\mathbb{P}_{\mathbf{y}\sim\mathrm{Ber}(\alpha)^n}(V(\mathbf{y}) = 1)$.

Fix any $V \in \mathcal{V}$ such that $\mathbb{E}(V) > \epsilon$. Since V is a conjunction of literals, we can write V in the form

$$V(\mathbf{y}) = V_0(\mathbf{y}_0) \cdot V_1(\mathbf{y}_1) \cdots V_{R-1}(\mathbf{y}_{R-1})$$

for some functions $V_0, V_1, \ldots, V_{R-1}\colon \{0,1\} \to \{0,1\}$. For each $0 \le a \le b \le R-1$, define

$$V_{a\cdots b} = V_a \cdot V_{a+1} \cdots V_b.$$

We inductively define numbers $0 = \ell_0 \le \ell_1 \le \cdots \le \ell_T$ as follows. Assume that we have already defined $\ell_0, \ldots, \ell_i$. Let ℓ_{i+1} be the smallest integer in $\{\ell_i + 1, \ldots, R-1\}$ such that

$$\mathbb{E}(V_{\ell_i \cdots \ell_{i+1}-1}) \le \epsilon^{1/T},$$

or let $\ell_{i+1} = R$ if no such ℓ_{i+1} exists. Define $V^{(i)} = V_{\ell_i \cdots \ell_{i+1}-1}$. Observe that $\ell_T = R$, because otherwise we would have

$$\epsilon < \mathbb{E}(V) \le \prod_{i=0}^{T-1} \mathbb{E}(V_i) \le (\epsilon^{1/T})^T = \epsilon,$$

a contradiction. Furthermore, $\mathbb{E}(V_i) > \epsilon^{1/T}/R = 2\epsilon_0$, because each literal in V is satisfied with probability at least $\min\{\alpha, 1-\alpha\} \ge 1/R$. Therefore, if we define

$$S_i = \{\mathbf{u} \in \{0,1\}^{r_0} : V_i(G_0(\mathbf{u})_{\ell_i \cdots \ell_{i+1}-1}) = 1\}$$

and $\rho_i = |S_i|/2^{r_0}$, then the correctness of G_0 ensures that $\rho_i \ge \epsilon_0$.

Next, we will show that there exist $\mathrm{hash}, \mathbf{w}^0, \ldots, \mathbf{w}^{T-1}$ such that for every i, we have $\mathrm{hash}(\mathbf{w}^i) \in S_i$. To prove it, pick $\mathrm{hash} \in \mathcal{H}$ at random. For each $i \in \{0, 1, \ldots, T-1\}$ and each $\mathbf{w} \in \{0,1\}^a$, let $X_{i,\mathbf{w}}$ be the indicator random variable for the "good" event $\mathrm{hash}(\mathbf{w}) \in S_i$. Define $X_i = \sum_{\mathbf{w}\in\{0,1\}^a} X_{i,\mathbf{w}}$. Then for every i, by pairwise independence, we have

$$\begin{aligned} \mathbb{E}(X_i) &= 2^a \cdot \rho_i \\ \text{and } \mathrm{Var}(X_i) &= 2^a \cdot \rho_i \cdot (1-\rho_i) \le 2^a \cdot \rho_i. \end{aligned}$$

Therefore, by Chebyshev's inequality,

$$\mathbb{P}(X_i = 0) \le \frac{1}{2^a \cdot \rho_i} \le \frac{1}{2^a \cdot \epsilon_0} < \frac{1}{R}.$$

Consequently, by the union bound over all i, there is a nonzero chance that $X_0 = X_1 = \cdots = X_{T-1} = 0$, in which case there exist $\mathbf{w}^0, \ldots, \mathbf{w}^{T-1}$ such that $\mathrm{hash}(\mathbf{w}^i) \in S_i$ for every i.

At this point, we have constructed our seed $\sigma = (\mathrm{hash}, \mathbf{w}^0, \ldots, \mathbf{w}^{T-1}, \ell_0, \ldots, \ell_T)$. By the construction of G, we have $V(G(\sigma)) = 1$. ■

Theorem 46 readily follows from Theorem 45, as we now explain.

Theorem 46 *Let $f\colon \{0,1\}^{d_0} \to \{0,1,\star\}$ be any function.*[7] *Let $N = |f^{-1}(\{0,1\})|$ and $N_1 = |f^{-1}(1)|$. There exists a depth-14 binary threshold network $\tilde{h}\colon \{0,1\}^{d_0} \to \{0,1\}$, with widths $\underline{\tilde{d}}$, satisfying the following.*

1. *$\tilde{h}$ is consistent with f, i.e., for every $\mathbf{x} \in \{0,1\}^{d_0}$, if $f(\mathbf{x}) \in \{0,1\}$, then $\tilde{h}(\mathbf{x}) = f(\mathbf{x})$.*
2. *The total number of weights in $\tilde{h}$ is at most $(1+o(1)) \cdot \log \binom{N}{N_1} + \mathrm{poly}(d_0)$. More precisely,*

$$w\left(\underline{\tilde{d}}\right) = \log \binom{N}{N_1} + \left(\log \binom{N}{N_1}\right)^{3/4} \cdot \mathrm{polylog}\, N + O(d_0^2 \cdot \log N)\,.$$

3. *Every layer of $\tilde{h}$ has width at most $(\log \binom{N}{N_1})^{3/4} \cdot \mathrm{poly}(d_0)$. More precisely,*

$$\underline{\tilde{d}}_{\max} = \left(\log \binom{N}{N_1}\right)^{3/4} \cdot \mathrm{polylog}\, N + O(d_0 \cdot \log N)\,.$$

Proof Let $R = 2^{2\lceil \log N \rceil}$. Let $C_0\colon \{0,1\}^{d_0} \to \{0,1\}^{\log R}$ be an $\mathbb{F}_2$-affine function that is injective on $\mathcal{X}$; such a function is guaranteed to exist by Lemma 27. Define $V\colon \{0,1\}^R \to \{0,1\}$ by the rule

$$V(\mathbf{y}) = 1 \iff \forall \mathbf{x} \in \mathcal{X},\ \mathbf{y}_{C_0(\mathbf{x})} = f(x).$$

This function V is a conjunction of N_1 variables and $N - N_1$ negated variables.

If $N_1 \in \{0, N\}$, then the theorem is trivial, because we can take $\tilde{h}$ to be a constant function. Assume, therefore, that $0 < N_1 < N$. Let $\alpha = N_1/N$, and note that $1/R \le \alpha \le 1 - 1/R$. Let $\epsilon = \frac{1}{2}\alpha^{N_1} \cdot (1-\alpha)^{N-N_1} = 2^{-H(\alpha)\cdot N - 1}$, and note that

$$\mathbb{P}_{\mathbf{y} \sim \mathrm{Ber}(\alpha)^R}(V(\mathbf{y}) = 1) = 2\epsilon.$$

Let $G\colon \{0,1\}^r \to \{0,1\}^R$ be the HSG from Theorem 45. There exists a seed $\sigma \in \{0,1\}^r$ such that $V(G(\sigma)) = 1$. Our network $\tilde{h}$ computes the function $\tilde{h}(\mathbf{x}) = G(\sigma)_{C_0(\mathbf{x})}$. Since $V(G(\sigma)) = 1$, we must have $\tilde{h}(\mathbf{x}) = f(\mathbf{x})$ for every $x \in \mathcal{X}$.

To bound the computational complexity, observe that $\log(1/\epsilon) = H(\alpha) \cdot N + 1 \le \log \binom{N}{N_1} + O(\log N)$. Therefore, every layer of $\tilde{h}$ has width at most

$$\left(\log \binom{N}{N_1}\right)^{3/4} \cdot \mathrm{polylog}\, N + O(d_0 \cdot \log N),$$

and the total number of weights in $\tilde{h}$ is at most

$$\log \binom{N}{N_1} + \left(\log \binom{N}{N_1}\right)^{3/4} \cdot \mathrm{polylog}\, N + O(d_0^2 \cdot \log N + d_0 \cdot \log^2 N).$$

Finally, we have $N \le 2^{d_0}$, so the last term above can be simplified to $O(d_0^2 \cdot \log N)$. ■

Remark 47 *In Theorem 46, the weights bound has an $O(d_0^2 \cdot \log N)$ term. This term is close to optimal; see Appendix F for further details.*

7. When $f(\mathbf{x}) = \star$, the interpretation is that f is "undefined" on $\mathbf{x}$, *i.e.,* f is a "partial" function.

E.6. XOR networks

In what follows, we denote the activation function $\sigma(x) = \mathbb{I}\{x > 0\}$.

Lemma 48 (XOR NN) *The* XOR *function can be implemented with a single-hidden-layer fully connected binary threshold network with input dimension 2 and* c_{XOR} *parameters by*

$$h_{\mathrm{XOR}}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \sigma\left(1 \odot \begin{pmatrix} 1 & 1 \end{pmatrix} \cdot \sigma\left(\begin{pmatrix} 1 \\ -1 \end{pmatrix} \odot \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 0 \\ 2 \end{pmatrix}\right) - 1\right).$$

Proof We can simplify h_{XOR} as

$$\begin{aligned}
h_{\mathrm{XOR}}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} &= \sigma\left(\begin{pmatrix} 1 & 1 \end{pmatrix} \cdot \sigma\left(\begin{pmatrix} 1 \\ -1 \end{pmatrix} \odot \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 0 \\ 2 \end{pmatrix}\right) - 1\right) \\
&= \sigma\left(\begin{pmatrix} 1 & 1 \end{pmatrix} \cdot \sigma\left(\begin{pmatrix} x_1 + x_2 \\ -x_1 - x_2 + 2 \end{pmatrix}\right) - 1\right) \\
&= \sigma\left(\sigma\left(x_1 + x_2\right) + \sigma\left(2 - x_1 - x_2\right) - 1\right) \\
&= \mathbb{I}\left\{\mathbb{I}\left\{x_1 + x_2 > 0\right\} + \mathbb{I}\left\{x_1 + x_2 < 2\right\} > 1\right\} \\
&= \mathbb{I}\left\{\mathbb{I}\left\{\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \neq \begin{pmatrix} 0 \\ 0 \end{pmatrix}\right\} + \mathbb{I}\left\{\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \neq \begin{pmatrix} 1 \\ 1 \end{pmatrix}\right\} > 1\right\} \\
&= \mathbb{I}\left\{\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \neq \begin{pmatrix} 0 \\ 0 \end{pmatrix} \text{ and } \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \neq \begin{pmatrix} 1 \\ 1 \end{pmatrix}\right\} \\
&= \mathrm{XOR}\left(x_1, x_2\right).
\end{aligned}$$

■

Remark 49 *Notice that the function* $\mathrm{Id} : \{0,1\} \to \{0,1\}$ *defined as* $\mathrm{Id}(0) = 0$ *and* $\mathrm{Id}(1) = 1$ *can be implemented using any depth* L *network with a single input dimension and* $c_{\mathrm{Id}} \cdot L$ *parameters.*

Following this remark, for simplicity we shall assume that h_1 and h_2 in the following Lemma are of the same depth, as they can be elongated with $O(L)$ additional parameters, which are negligible in the subsequent analysis.

Lemma 50 *Let h_1, h_2 be two binary networks with depths $L_1 \leq L_2$ and widths $\underline{d}^{(1)}, \underline{d}^{(2)}$, respectively. Then, there exists a network h with depth $L_{\text{XOR}} \triangleq L_2 + 2$ and widths*

$$\underline{d}_{\text{XOR}} \triangleq \left(d_1^{(1)} + d_1^{(2)},\, \ldots,\, d_{L_1}^{(1)} + d_{L_1}^{(2)},\, d_{L_1+1}^{(2)} + 1,\, \ldots,\, d_{L_2}^{(2)} + 1,\, 2,\, 1\right),$$

such that for all inputs $\mathbf{x} \in \{0,1\}^{d_0}$, $h(\mathbf{x}) = h_1(\mathbf{x}) \oplus h_2(\mathbf{x})$.

The lemma above is given immediately by the lemma we state and prove next.

Lemma 51 (XOR of Two NNs) *Let $h_1, h_2 : \mathcal{X} \to \{0,1\}$ be quantized fully connected binary threshold networks with depths L' and widths $\underline{d}^{(1)}, \underline{d}^{(2)}$, respectively. Let $L \geq 2 + L'$ and $\underline{d} \geq \underline{d}_{\text{XOR}}$. Let $\Theta^{BTN}(\underline{d}; h_1, h_2)$ be the subset of $\Theta^{BTN}(\underline{d})$ such that for all $\boldsymbol{\theta} \in \Theta^{BTN}(\underline{d}; h_1, h_2)$, $\boldsymbol{\theta}$ has the following form:*

- *For $l = 1$:*

$$\mathbf{W}_1 = \begin{pmatrix} \mathbf{W}_1^{(1)} \\ \mathbf{W}_1^{(2)} \\ \tilde{\mathbf{W}}_1 \end{pmatrix},\ \mathbf{b}_1 = \begin{pmatrix} \mathbf{b}_1^{(1)} \\ \mathbf{b}_1^{(2)} \\ \tilde{\mathbf{b}}_1 \end{pmatrix},\ \gamma_1 = \begin{pmatrix} \mathbf{1}_{d_l^{(1)}} \\ \mathbf{1}_{d_l^{(2)}} \\ \mathbf{0}_{d_l - d_l^{(1)} - d_l^{(2)}} \end{pmatrix}$$

  with arbitrary $\tilde{\mathbf{W}}_1, \tilde{\mathbf{b}}_1$.

- *For $l = 2, \ldots, L'$:*

$$\mathbf{W}_l = \begin{pmatrix} \mathbf{W}_l^{(1)} & \mathbf{0}_{d_l^{(1)} \times d_{l-1}^{(2)}} & \tilde{\mathbf{W}}_l^1 \\ \mathbf{0}_{d_l^{(2)} \times d_{l-1}^{(1)}} & \mathbf{W}_l^{(2)} & \tilde{\mathbf{W}}_l^2 \\ \tilde{\mathbf{W}}_l^3 & \tilde{\mathbf{W}}_l^4 & \tilde{\mathbf{W}}_l^5 \end{pmatrix} \in \{0,1\}^{d_l \times d_{l-1}},$$

$$\mathbf{b}_l = \begin{pmatrix} \mathbf{b}_l^{(1)} \\ \mathbf{b}_l^{(2)} \\ \tilde{\mathbf{b}}_l \end{pmatrix} \in \{-d_{l-1}, \ldots, -1, 0, 1, \ldots, d_{l-1} - 1\}^{d_l},$$

$$\gamma_l = \begin{pmatrix} \mathbf{1}_{d_l^{(1)}} \\ \mathbf{1}_{d_l^{(2)}} \\ \mathbf{0}_{d_l - d_l^{(1)} - d_l^{(2)}} \end{pmatrix} \in \{0,1\}^{d_l},$$

  with arbitrary $\tilde{\mathbf{W}}_l^1, \tilde{\mathbf{W}}_l^2, \tilde{\mathbf{W}}_l^3, \tilde{\mathbf{W}}_l^4, \tilde{\mathbf{W}}_l^5, \tilde{\mathbf{b}}_l$.

- *For $l = L' + k$, $k = 1, 2$:*

$$\mathbf{W}_l = \begin{pmatrix} \mathbf{W}_k^{\text{XOR}} & \tilde{\mathbf{W}}_l^1 \\ \tilde{\mathbf{W}}_l^2 & \tilde{\mathbf{W}}_l^3 \end{pmatrix} \in \{0,1\}^{d_l \times d_{l-1}},$$

$$\mathbf{b}_l = \begin{pmatrix} \mathbf{b}_k^{\text{XOR}} \\ \tilde{\mathbf{b}}_l \end{pmatrix} \in \{-d_{l-1}, \ldots, -1, 0, 1, \ldots, d_{l-1} - 1\}^{d_l}, \gamma_l = \begin{pmatrix} \gamma_k^{\text{XOR}} \\ \mathbf{0} \end{pmatrix} \in \{0, \pm 1\}^{d_l}.$$

- *And for $l > L' + 2$:*

$$\mathbf{W}_l = \begin{pmatrix} \mathbf{W}^{Id} & \tilde{\mathbf{W}}_l^1 \\ \tilde{\mathbf{W}}_l^2 & \tilde{\mathbf{W}}_l^3 \end{pmatrix} \in \{0,1\}^{d_l \times d_{l-1}},$$
$$\mathbf{b}_l = \begin{pmatrix} \mathbf{b}^{Id} \\ \tilde{\mathbf{b}}_l \end{pmatrix} \in \{-d_{l-1}, \dots, -1, 0, 1, \dots, d_{l-1} - 1\}^{d_l}, \gamma_l = \begin{pmatrix} \gamma^{Id} \\ \mathbf{0} \end{pmatrix} \in \{0, \pm 1\}^{d_l}.$$

Then for all $\boldsymbol{\theta} \in \Theta^{BTN}(\underline{d}; h_1, h_2)$ $h_{\boldsymbol{\theta}} = h_1 \oplus h_2$.

An illustration of this construction is given in Figure 3.

Proof We prove the claim by induction. For $l = 1$ we have $d_0 = d_0^{(1)} = d_0^{(2)}$ and

$$\begin{aligned} h_\theta^{(1)}(\mathbf{x}) &= \gamma_1 \odot \sigma\left(\mathbf{W}_1 h_\theta^{(0)}(\mathbf{x}) + \mathbf{b}_1\right) \\ &= \begin{pmatrix} \mathbf{1}_{d_1^\star} \\ \mathbf{1}_{d_1^f} \\ \mathbf{0}_{d_1 - d_1^\star - d_1^f} \end{pmatrix} \odot \sigma\left(\begin{pmatrix} \mathbf{W}_1^{(1)} \\ \mathbf{W}_1^{(2)} \\ \tilde{\mathbf{W}}_1^1 \end{pmatrix} \mathbf{x} + \begin{pmatrix} \mathbf{b}_1^{(1)} \\ \mathbf{b}_1^{(2)} \\ \tilde{\mathbf{b}}_1 \end{pmatrix}\right) \\ &= \begin{pmatrix} \sigma\left(\mathbf{W}_1^{(1)}\mathbf{x} + \mathbf{b}_1^{(1)}\right) \\ \sigma\left(\mathbf{W}_1^{(2)}\mathbf{x} + \mathbf{b}_1^{(2)}\right) \\ \mathbf{0}_{d_1 - d_1^{(1)} - d_1^{(2)}} \end{pmatrix} = \begin{pmatrix} h_1^{(1)}(\mathbf{x}) \\ h_2^{(1)}(\mathbf{x}) \\ \mathbf{0}_{d_1 - d_1^\star - d_1^f} \end{pmatrix}. \end{aligned}$$

Assume that for some $l \leq L'$ we have

$$h_\theta^{(l-1)}(\mathbf{x}) = \begin{pmatrix} h_1^{(l-1)}(\mathbf{x}) \\ h_2^{(l-1)}(\mathbf{x}) \\ \mathbf{0}_{d_l - d_l^\star - d_l^f} \end{pmatrix}.$$

Then,

$$\begin{aligned} h_\theta^{(l)}(\mathbf{x}) &= \gamma_l \odot \sigma\left(\mathbf{W}_l h_\theta^{(l-1)}(\mathbf{x}) + \mathbf{b}_l\right) \\ &= \begin{pmatrix} \mathbf{1}_{d_l^{(1)}} \\ \mathbf{1}_{d_l^{(2)}} \\ \mathbf{0}_{d_l - d_l^{(1)} - d_l^{(2)}} \end{pmatrix} \odot \sigma\left(\begin{pmatrix} \mathbf{W}_l^{(1)} & \mathbf{0}_{d_l^{(1)} \times d_{l-1}^{(2)}} & \tilde{\mathbf{W}}_l^1 \\ \mathbf{0}_{d_l^{(2)} \times d_{l-1}^{(1)}} & \mathbf{W}_l^{(2)} & \tilde{\mathbf{W}}_l^2 \\ \tilde{\mathbf{W}}_l^3 & \tilde{\mathbf{W}}_l^4 & \tilde{\mathbf{W}}_l^5 \end{pmatrix} \begin{pmatrix} h_1^{(l-1)}(\mathbf{x}) \\ h_2^{(l-1)}(\mathbf{x}) \\ \mathbf{0}_{d_l - d_l^{(1)} - d_l^{(2)}} \end{pmatrix} + \begin{pmatrix} \mathbf{b}_l^{(1)} \\ \mathbf{b}_l^{(2)} \\ \tilde{\mathbf{b}}_l \end{pmatrix}\right) \\ &= \begin{pmatrix} \sigma\left(\mathbf{W}_l^{(1)} h_1^{(l-1)}(\mathbf{x}) + \mathbf{b}_l^{(1)}\right) \\ \sigma\left(\mathbf{W}_l^{(2)} h_2^{(l-1)}(\mathbf{x}) + \mathbf{b}_l^{(2)}\right) \\ \mathbf{0}_{d_l - d_l^{(1)} - d_l^{(2)}} \end{pmatrix} = \begin{pmatrix} h_1^{(l)}(\mathbf{x}) \\ h_2^{(l)}(\mathbf{x}) \\ \mathbf{0}_{d_l - d_l^{(1)} - d_l^{(2)}} \end{pmatrix}. \end{aligned}$$

It is left to show that the claim holds for $l > L'$. By the previous steps, $h_\theta^{(L')}(\mathbf{x}) = \begin{pmatrix} h_1(\mathbf{x}) \\ h_2(\mathbf{x}) \\ \mathbf{0}_{d_{L'} - 2} \end{pmatrix}$.

Under the assumptions on $\mathbf{W}_{L'+k}, \mathbf{b}_{L'+k}$ and $\gamma_{L'+k}$, $k = 1, 2$ it holds that

$$h_\theta^{(L'+2)}(\mathbf{x}) = \begin{pmatrix} h_1(\mathbf{x}) \oplus h_2(\mathbf{x}) \\ \mathbf{0}_{d_{L'} - 1} \end{pmatrix}.$$

Under the assumptions on layers $l > L' + 2$,

$$h_\theta^{(l)}(\mathbf{x}) = \begin{pmatrix} h_1(\mathbf{x}) \oplus h_2(\mathbf{x}) \\ \mathbf{0}_{d_l - 1} \end{pmatrix}.$$

In particular, assuming that $d_L = 1$, $h_\theta(\mathbf{x}) = h_1(\mathbf{x}) \oplus h_2(\mathbf{x})$. ■

Corollary 52 *Let h_1, h_2 be networks with depths L_1, L_2 and widths $\underline{d}^{(1)}, \underline{d}^{(2)}$. Then $h_1 \oplus h_2$ can be implemented with a network h of depth $L = \max\{L_1, L_2\} + 2$ and widths $\underline{d}$ such that*

$$w(\underline{d}) \leq w\left(\underline{d}^{(1)}\right) + w\left(\underline{d}^{(2)}\right) + 2\underline{d}^{(2)}_{\max} \cdot n\left(\underline{d}^{(1)}\right) + O(1)$$

and

$$\underline{d}_{\max} \leq \underline{d}^{(1)}_{\max} + \underline{d}^{(2)}_{\max}.$$

Proof Following 49 we assume shall assume that $L_1 = L_2 = L$. We know from 51 that there exists a network h with dimensions $\underline{d} = \left(\underline{d}^{(1)} + \underline{d}^{(2)}, 2, 1\right)$ such that $h = h_1 \oplus h_2$. Therefore

$$\begin{aligned}
w(\underline{d}) &= \left(d_1^{(1)} + d_1^{(2)}\right) d_0 + \sum_{l=2}^{L} \left(d_l^{(1)} + d_l^{(2)}\right)\left(d_{l-1}^{(1)} + d_{l-1}^{(2)}\right) + O(1) \\
&= d_1^{(1)} d_0 + \sum_{l=2}^{L} d_l^{(1)} d_{l-1}^{(1)} + d_1^{(2)} d_0 + \sum_{l=2}^{L} d_l^{(2)} d_{l-1}^{(2)} + \sum_{l=2}^{L} \left[d_l^{(1)} d_{l-1}^{(2)} + d_l^{(2)} d_{l-1}^{(1)}\right] + O(1) \\
&= w\left(\underline{d}^{(1)}\right) + w\left(\underline{d}^{(2)}\right) + \sum_{l=2}^{L} \left[d_l^{(1)} d_{l-1}^{(2)} + d_l^{(2)} d_{l-1}^{(1)}\right] + O(1) \\
&\leq w\left(\underline{d}^{(1)}\right) + w\left(\underline{d}^{(2)}\right) + \sum_{l=2}^{L} \left[d_l^{(1)} \underline{d}^{(2)}_{\max} + \underline{d}^{(2)}_{\max} d_{l-1}^{(1)}\right] + O(1) \\
&= w\left(\underline{d}^{(1)}\right) + w\left(\underline{d}^{(2)}\right) + \underline{d}^{(2)}_{\max} \sum_{l=2}^{L} \left[d_l^{(1)} + d_{l-1}^{(1)}\right] + O(1) \\
&\leq w\left(\underline{d}^{(1)}\right) + w\left(\underline{d}^{(2)}\right) + 2\underline{d}^{(2)}_{\max} \cdot n\left(\underline{d}^{(1)}\right) + O(1).
\end{aligned}$$

In addition, $\underline{d}_{\max} \leq \underline{d}^{(1)}_{\max} + \underline{d}^{(2)}_{\max}$ and $n(\underline{d}) = n\left(\underline{d}^{(1)}\right) + n\left(\underline{d}^{(2)}\right)$. ■

Recall Corollary 4. For any teacher $h^\star$ of depth $L^\star$ and dimensions $\underline{d}^\star$ and any consistent training set S generated from it, there exists an interpolating network h (*i.e.,* $\mathcal{L}_S(h) = 0$) of depth $L = \max\{L^\star, 14\} + 2$ and dimensions $\underline{d}$, such that the number of edges is

$$w(\underline{d}) \leq w(\underline{d}^\star) + N \cdot H(\mathcal{L}_S(h^\star)) + 2n(\underline{d}^\star) N^{3/4} \mathrm{polylog} N + O(d_0(d_0 + n(\underline{d}^\star)) \cdot \log N)$$

and the dimensions are

$$\underline{d}_{\max} \leq \underline{d}^\star_{\max} + N^{3/4} \cdot H(\mathcal{L}_S(h^\star)) \cdot \mathrm{polylog}(N) + O(d_0 \cdot \log(N)).$$

Proof We use Corollary 52 with $h_1 = h^\star$ and $h_2 = \tilde{h}_S$, the noise memorizing network from Theorem 46, to get

$$\begin{aligned}
w(\underline{d}) &\leq w(\underline{d}^\star) + w\left(\tilde{\underline{d}}_S\right) + 2\tilde{\underline{d}}_{S,\max} \cdot n(\underline{d}^\star) + O(1) \\
&\leq w(\underline{d}^\star) + \log\binom{N}{N_1} + \left(\log\binom{N}{N_1}\right)^{3/4} \cdot \mathrm{polylog}\, N + O(d_0^2 \cdot \log N) \\
&\quad + 2n(\underline{d}^\star)\left(\left(\log\binom{N}{N_1}\right)^{3/4} \cdot \mathrm{polylog}\, N + O(d_0 \cdot \log N)\right) + O(1).
\end{aligned}$$

Using Stirling's approximation

$$\log\binom{N}{N_1} = N \cdot H\left(\frac{N_1}{N}\right) + O\left(\log\left(N\right)\right) = N \cdot H\left(\mathcal{L}_S\left(h^\star\right)\right) + O\left(\log\left(N\right)\right) .$$

Therefore

$$\begin{aligned} w\left(\underline{d}\right) &\leq w\left(\underline{d}^\star\right) + N \cdot H\left(\mathcal{L}_S\left(h^\star\right)\right) + O\left(\log\left(N\right)\right) + N^{3/4} \cdot \mathrm{polylog}\, N + O\left(d_0^2 \cdot \log N\right) \\ &\quad + 2n\left(\underline{d}^\star\right)\left(N^{3/4} \cdot \mathrm{polylog}\, N + O\left(d_0 \cdot \log N\right)\right) \\ &= w\left(\underline{d}^\star\right) + N \cdot H\left(\mathcal{L}_S\left(h^\star\right)\right) + 2n\left(\underline{d}^\star\right) N^{3/4} \mathrm{polylog} N + O\left(d_0\left(d_0 + n\left(\underline{d}^\star\right)\right) \cdot \log N\right) . \end{aligned}$$

The bound of $\underline{d}_{\max}$ is derived similarly. ■

Appendix F. The label-flip-memorization network's dependence on the dimension

In Theorem 46, the wire bound has an $O(d_0^2 \cdot \log N)$ term. (Recall that d_0 is the input dimension and N is the domain size.) In this section, we discuss (a) approaches for improving this term and (b) a lower bound showing that it cannot be significantly improved.

F.1. Improving the $O(d_0^2 \cdot \log N)$ Term

The $O(d_0^2 \cdot \log N)$ term in Theorem 46 can be improved by using the following fact.

Lemma 53 (Using a sign matrix for preprocessing) *Let $d_0 \in \mathbb{N}$, let $\hat{\mathcal{X}} \subseteq \{0,1\}^{d_0}$, and let $N = |\hat{\mathcal{X}}|$. There exists $d_1 = O(\sqrt{d_0} \cdot \log N)$ and there exists a matrix $\mathbf{W} \in \{\pm 1\}^{d_1 \times d_0}$ such that the function $C_0 \colon \{0,1\}^{d_0} \to \{0,1\}^{d_1}$ defined by $C_0(\mathbf{x}) = \mathbb{I}\{\mathbf{W}\mathbf{x} > 0\}$ is injective on $\hat{\mathcal{X}}$.*

Proof Pick $\mathbf{W} \in \{\pm 1\}^{d_1 \times d_0}$ uniformly at random. We will show that there is a nonzero chance that C_0 is injective on $\hat{\mathcal{X}}$.

Let $\mathbf{x}, \mathbf{x}'$ be any two distinct points in $\hat{\mathcal{X}}$. Consider a single row $\mathbf{W}_i$ of $\mathbf{W}$. Let E be the good event that

$$\mathbf{W}_i \cdot (\mathbf{x} \odot \mathbf{x}') \in \{0,1\}.$$

Then $\Pr[E] \geq \Omega(1/\sqrt{d_0})$, because we are taking a simple one-dimensional random walk of length at most d_0. Conditioned on E, there is an $\Omega(1)$ chance that $\mathbb{I}\{\mathbf{W}_i \cdot \mathbf{x} > 0\} \neq \mathbb{I}\{\mathbf{W}_i \cdot \mathbf{x}' > 0\}$, because we are taking two independent one-dimensional random walks starting from either 0 or 1, at least one of which has nonzero length, and asking whether they land on the same side of $1/2$. Therefore, unconditionally, $\Pr[\mathbb{I}\{\mathbf{W}_i \cdot \mathbf{x} > 0\} \neq \mathbb{I}\{\mathbf{W}_i \cdot \mathbf{x}' > 0\}] \geq \Omega(1/\sqrt{d_0})$. Consequently, by independence,

$$\Pr[C_0(\mathbf{x}) = C_0(\mathbf{x}')] \leq (1 - \Omega(1/\sqrt{d_0}))^{d_1} < 1/N^2,$$

provided we choose a suitable value $d_1 = O(\sqrt{d_0} \cdot \log N)$. By the union bound over all pairs $\mathbf{x}, \mathbf{x}'$, it follows that there is a nonzero chance that C_0 is injective on $\hat{\mathcal{X}}$. ■

There are two approaches to using Lemma 53 for the sake of improving the $O(d_0^2 \cdot \log N)$ term in Theorem 46.

- One approach would be to start with a trivial layer that copies the input $\mathbf{x} \in \{0,1\}^{d_0}$ as well as computing all the negations of the bits of $\mathbf{x}$; then we have a layer that applies the function C_0 from Lemma 53 (using negated variables to implement -1 weights); and then we continue with the network of Theorem 46. The net effect is that the depth has increased by two (so the network now has depth 16 instead of 14), and in the weights bound, the $O(d_0^2 \cdot \log N)$ term has been slightly improved to $O(d_0^2 + d_0^{3/2} \cdot \log N + d_0 \cdot \log^3 N)$.
- A second approach would be to change the model. If we permit ternary edge weights (i.e., weights in the set $\{-1, 0, 1\}$), then the function C_0 of Lemma 53 can be implemented as the very first layer of our network, and then we can continue with the network of Theorem 46. Note that we need ternary edge weights only in the first layer; the edge weights in all subsequent layers are binary. The benefit of this approach is in the weights bound, the $O(d_0^2 \cdot \log N)$ term of Theorem 46 would be improved to $O(d_0^{3/2} \cdot \log N + d_0 \cdot \log^3 N)$.

F.2. A d_0^2 Lower Bound on the Number of Weights

We now show that the $O(d_0^2 \cdot \log N)$ term in Theorem 46 cannot be improved to something better than d_0^2, if we insist on using the "binary threshold network" model. The argument is elementary.

Proposition 54 (d_0^2 wire lower bound) *For every $d_0 \in \mathbb{N}$, there exists a partial Boolean function $f\colon \{0,1\}^{d_0} \to \{0,1,\star\}$, defined on a domain $\hat{\mathcal{X}}$ of size d_0+1, such that for every binary threshold network $\tilde{h}$, if $\tilde{h}$ agrees with f everywhere in its domain and $\underline{d}$ is the widths of $\tilde{h}$, then $w(\underline{d}) \geq d_0^2$.*

Proof For each $i \in \{0,1,\ldots,d_0\}$, let $\mathbf{x}^{(i)}$ be the vector consisting of i zeroes followed by $d_0 - i$ ones. Let $\hat{\mathcal{X}} = \{\mathbf{x}^{(i)} : 0 \leq i \leq d_0\}$, and let

$$f(\mathbf{x}) = \begin{cases} \mathsf{PARITY}(\mathbf{x}) & \text{if } \mathbf{x} \in \hat{\mathcal{X}} \\ \star & \text{otherwise.} \end{cases}$$

For the analysis, let $\tilde{h}$ be a fully connected binary threshold network that agrees with f on all points in $\hat{\mathcal{X}}$. Consider the layer immediately following the input layer. Each node g in this layer computes either a monotone Boolean function or an anti-monotone Boolean function of the input variables. Therefore, there is at most one value $i \in \{1,2,\ldots,d_0\}$ such that $g(\mathbf{x}^{(i-1)}) \neq g(\mathbf{x}^{(i)})$. On the other hand, for every $i \in \{1,2,\ldots,d_0\}$, we have $\tilde{h}(\mathbf{x}^{(i-1)}) \neq \tilde{h}(\mathbf{x}^{(i)})$, and hence there must be at least one node g in this layer such that $g(\mathbf{x}^{(i-1)}) \neq g(\mathbf{x}^{(i)})$. Therefore, there are at least d_0 many nodes g.

Thus, the first two layers of $\tilde{h}$ both have widths of at least d_0, demonstrating that $\tilde{h}$ has at least d_0^2 many weights. ■

Appendix G. Generalization results (Proofs for Section 4)

Denote by $\mathcal{H}_{\underline{d}}^{\mathrm{BTN}}$ the set of functions representable as binary threshold networks with dimensions $\underline{d}$ (given a fixed depth L). We start by bounding the cardinality $\left|\mathcal{H}_{\underline{d}}^{\mathrm{BTN}}\right|$ in terms of the number of edges $w\left(\underline{d}\right)$.

Lemma 55 *Let $\underline{d}$ be the dimensions of a binary threshold network with $w \triangleq w\left(\underline{d}\right)$ edges. Then there are $2^{w+O\left(\sqrt{w}\log(w)\right)}$ functions representable as networks with dimensions $\underline{d}$.*

Proof We bound the number of function representable as binary threshold networks with dimensions $\underline{d}$ having w edges by suggesting a way to encode them, and then bounding the number of bits in the encoding. First, permute each layer so the neurons are sorted by the bias and neuron scaling terms (b_{li}, γ_{li}). As NNs are invariant to permutations, this does not change the function. Now, at each layer we encode the bias term based on one of two encodings.

- If $d_l < d_{l-1}$, then list each of the bias terms as a number with $O\left(\log\left(d_{l-1}\right)\right)$ bits plus 2 bits for the scaling term for a total of $O\left(d_l\left(\log\left(d_{l-1}\right)+2\right)\right) \leq O\left(\sqrt{d_l d_{l-1}}\log\left(d_{l-1}\right)\right)$, where the inequality is due to $d_l < d_{l-1}$.
- If $d_l \geq d_{l-1}$, then we encode the bias and scaling terms by listing the number of times each pair $(b_{li}, \gamma_{li}) \in \{-d_{l-1}, \ldots, d_{l-1}-1\} \times \{-1, 0, 1\}$ appears in $(\mathbf{b}_l, \boldsymbol{\gamma_l})$ (recall that the neurons are ordered according to these pairs). Each pair can appear at most d_l times and so requires $O\left(\log\left(d_l\right)\right)$ bits to encode for a total of $O\left(6d_{l-1}\log\left(d_l\right)\right) = O\left(d_{l-1}\log\left(d_l\right)\right) \leq O\left(\sqrt{d_l d_{l-1}}\log\left(d_l d_{l-1}\right)\right)$.

By encoding each weight with a single bit, this means that for all layers, we can encode the weights, biases and scaling terms using $d_l d_{l-1} + O\left(\sqrt{d_l d_{l-1}}\log\left(d_l d_{l-1}\right)\right)$ bits for a total of

$$
\begin{aligned}
\sum_{l=1}^{L} d_l d_{l-1} + O\left(\sqrt{d_l d_{l-1}}\log\left(d_l d_{l-1}\right)\right) &= w + O\left(\sum_{l=1}^{L}\sqrt{d_l d_{l-1}}\log\left(d_l d_{l-1}\right)\right) \\
&\leq w + O\left(\sum_{l=1}^{L}\sqrt{d_l d_{l-1}}\log\left(\sum_{l=1}^{L} d_l d_{l-1}\right)\right) \\
&\leq w + O\left(\sum_{l=1}^{L}\sqrt{d_l d_{l-1}}\log\left(w\right)\right) = w + O\left(\log\left(w\right)\cdot L\sum_{l=1}^{L}\frac{1}{L}\sqrt{d_l d_{l-1}}\right) \\
\text{[Jensen]} &\leq w + O\left(\log\left(w\right)\cdot L\sqrt{\sum_{l=1}^{L}\frac{1}{L}d_l d_{l-1}}\right) = w + O\left(\log\left(w\right)\cdot\sqrt{L}\sqrt{\sum_{l=1}^{L} d_l d_{l-1}}\right) \\
&= w + O\left(\log\left(w\right)\cdot\sqrt{L}\sqrt{w}\right) \\
&= w + O\left(\log\left(w\right)\cdot\sqrt{w}\right) .
\end{aligned}
$$

■

Corollary 56 *Assuming that the depth L is fixed and known, a binary threshold network of depth L with unknown number of weights w, can be encoded with $w + O\left(\sqrt{w}\log\left(w\right)\right)$ bits.*

Proof After specifying the architecture $\underline{d}$, from Lemma 55 we require $w + O\left(\sqrt{w}\log\left(w\right)\right)$ bits. Therefore it remains to bound the length of the encoding of $\underline{d}$. We first use $O\left(\log\left(w\right)\right)$ bits to encode the number of weights, then, since $\underline{d} \in [w]^L$, we only need $O\left(\log\left(w^L\right)\right) = O\left(\log\left(w\right)\right)$ additional bits for a total of $w + O\left(\sqrt{w}\log\left(w\right)\right) + O\left(\log\left(w\right)\right) = w + O\left(\sqrt{w}\log\left(w\right)\right)$. ■

G.1. Derivation of the min-size generalization bounds (Proofs for Section 4.1)

Throughout this subsection, we use $A(S)$ to denote the min-size interpolating NN of depth L, $A_L(S)$.

Lemma 57 *Let $L \geq 16$ be fixed. Then*

$$I(S; A(S)) \leq w(\underline{d}^\star) + N \cdot H(\varepsilon^\star) + O(\delta(N, d_0, \underline{d}^\star))$$

where

$$\delta(N, d_0, \underline{d}^\star) = n(\underline{d}^\star) \cdot N^{3/4} \cdot \text{polylog}(N + n(\underline{d}^\star) + d_0) + d_0^2 \cdot \log N + d_0 n(\underline{d}^\star) \log(n(\underline{d}^\star) + N + d_0)^{3/2}.$$

Proof Using Shannon's source coding theorem:

$$I(S; A(S)) \leq H(A(S)) \leq \mathbb{E}|A(S)|,$$

where $|A(S)|$ denotes the number of bits in the encoding of $A(S)$. Following Corollary 4, for a consistent S, $A(S)$ is a network with fixed depth and at most

$$w \triangleq w(\underline{d}^\star) + N \cdot H(\mathcal{L}_S(h^\star)) + 2n(\underline{d}^\star) N^{3/4} \text{polylog} N + O(d_0(d_0 + n(\underline{d}^\star)) \cdot \log N)$$

weights and therefore, using the result from Corollary 56 and $\sqrt{w(\underline{d}^\star)} \leq d_0 + n(\underline{d}^\star)$

$$\begin{aligned}
|A(S)| &\leq w + O\left(\sqrt{w} \log(w)\right) \\
&= w(\underline{d}^\star) + N \cdot H(\mathcal{L}_S(h^\star)) \\
&\quad + O\Big(n(\underline{d}^\star) \cdot N^{3/4} \cdot \text{polylog}(N + n(\underline{d}^\star) + d_0) + d_0^2 \cdot \log N + d_0 n(\underline{d}^\star) \log(n(\underline{d}^\star) + N + d_0)^{3/2}\Big) \\
&= w(\underline{d}^\star) + N \cdot H(\mathcal{L}_S(h^\star)) + O(\delta(N, d_0, \underline{d}^\star)).
\end{aligned}$$

In case S is inconsistent, $A(S) = \star$ so $|A(S)| = O(1)$. Taking the expected value and using Jensen's inequality gives

$$\begin{aligned}
\mathbb{E}|A(S)| &= \mathbb{E}[|A(S)| \cdot \mathbb{I}\{\text{inconsistent } S\}] + \mathbb{E}[|A(S)| \cdot \mathbb{I}\{\text{consistent } S\}] \\
&\leq O(1) + \mathbb{E}\Big[\underbrace{\mathbb{I}\{\text{consistent } S\}}_{\leq 1} \underbrace{(w(\underline{d}^\star) + N \cdot H(\mathcal{L}_S(h^\star)) + O(\delta(N, d_0, \underline{d}^\star)))}_{\geq 0}\Big] \\
&\leq O(1) + \mathbb{E}[w(\underline{d}^\star) + N \cdot H(\mathcal{L}_S(h^\star)) + O(\delta(N, d_0, \underline{d}^\star))] \\
&= O(1) + w(\underline{d}^\star) + O(\delta(N, d_0, \underline{d}^\star)) + N \cdot \mathbb{E}[H(\mathcal{L}_S(h^\star))] \\
\text{[Jensen]} &\leq w(\underline{d}^\star) + N \cdot H(\mathbb{E}[\mathcal{L}_S(h^\star)]) + O(\delta(N, d_0, \underline{d}^\star)) \\
&= w(\underline{d}^\star) + N \cdot H(\varepsilon^\star) + O(\delta(N, d_0, \underline{d}^\star)).
\end{aligned}$$

■

With this result, we are ready to derive the generalization results.

Recall Theorem 6. Consider a distribution $\mathcal{D}$ induced by a noisy teacher model of depth $L^\star$ and widths $\underline{d}^\star$ (Assumption 2) with a noise level of $\varepsilon^\star < 1/2$. Let $S \sim \mathcal{D}^N$ be a training set such that $N = o(\sqrt{1/\mathcal{D}_{\max}})$. Then, for any fixed depth $L \geq \max\{L^\star, 14\} + 2$, the generalization error of the min-size depth-L NN interpolator satisfies the following.

- **Under arbitrary label noise**,

$$\mathbb{E}_S\left[\mathcal{L}_{\mathcal{D}}\left(A\left(S\right)\right)\right] \leq 1 - 2^{-H(\varepsilon^\star)/\mathbb{P}_S(\text{consistent } S)} + \mathbb{P}\left(\text{inconsistent } S\right) + O\left(C_{\min}\left(N, d_0, \underline{d}^\star\right)\right) .$$

- **Under independent label noise,**

$$\begin{aligned}&\left|\mathbb{E}_S\left[\mathcal{L}_{\mathcal{D}}\left(A\left(S\right)\right)\right] - 2\varepsilon^\star\left(1-\varepsilon^\star\right)\right| \\ &\leq \left(1-2\varepsilon^\star\right)\sqrt{\frac{O(C_{\min}(N,d_0,\underline{d}^\star))+\mathbb{P}(\text{inconsistent } S)}{\mathbb{P}(\text{consistent } S)}} + \frac{\left(N-1\right)\mathcal{D}_{\max}}{3} + \mathbb{P}\left(\text{inconsistent } S\right) ,\end{aligned}$$

where

$$\begin{aligned}C_{\min}\left(N, d_0, \underline{d}^\star\right) &= \frac{w\left(\underline{d}^\star\right) + \delta\left(N, d_0, \underline{d}^\star\right)}{N} \\ &= \frac{w\left(\underline{d}^\star\right) + n\left(\underline{d}^\star\right)\cdot N^{3/4}\cdot\text{polylog}\left(N+n(\underline{d}^\star)+d_0\right)+d_0^2\cdot\log N+d_0 n\left(\underline{d}^\star\right)\log\left(n\left(\underline{d}^\star\right)+N+d_0\right)^{3/2}}{N}.\end{aligned}$$

Remark 58 *The bound shown in Section 4.1 is found by bounding $\mathbb{P}(\text{inconsistent } S) \leq \frac{1}{2}N^2\mathcal{D}_{\max}$ as in Lemma 18. Then using the Taylor approximation with small $N^2\mathcal{D}_{\max}$*

$$\begin{aligned}1 - 2^{-\frac{H(\varepsilon^\star)}{\mathbb{P}(\text{consistent } S)}} &\leq 1 - 2^{-\frac{H(\varepsilon^\star)}{1-\frac{1}{2}N^2\mathcal{D}_{\max}}} \\ &= 1 - 2^{-H(\varepsilon^\star)\left(1+O\left(N^2\mathcal{D}_{\max}\right)\right)} \\ &= 1 - 2^{-H(\varepsilon^\star)}\left(1 + O\left(N^2\mathcal{D}_{\max}\right)\right) \\ &= 1 - 2^{-H(\varepsilon^\star)} + O\left(N^2\mathcal{D}_{\max}\right) .\end{aligned}$$

Lemma 18 is used similarly to bound the error in the independent noise case.

Assuming that $N = \omega\left(n\left(\underline{d}^\star\right)^4\text{polylog}\left(n\left(\underline{d}^\star\right)\right) + d_0^2\log d_0\right)$ we can deduce that $N = \omega\left(w\left(\underline{d}^\star\right)\right)$ as well since

$$w\left(\underline{d}^\star\right) \leq \left(n\left(\underline{d}^\star\right) + d_0\right)^2 \leq 4\left(\max\left\{n\left(\underline{d}^\star\right), d_0\right\}\right)^2 .$$

Together with $N = o\left(\sqrt{1/\mathcal{D}_{\max}}\right)$ we get the desired form of the bounds.

Proof Starting with the bound in the arbitrary noise setting, we combine 21 with 57

$$\begin{aligned}-\log\left(1 - \mathbb{E}_S\left[\mathcal{L}_{\mathcal{D}}\left(A\left(S\right)\right) \mid \text{consistent } S\right]\right) &\leq \frac{I\left(S; A\left(S\right)\right)}{N\cdot\mathbb{P}_S\left(\text{consistent } S\right)} \\ &\leq \frac{w(\underline{d}^\star)+N\cdot H(\varepsilon^\star)+O\left(n(\underline{d}^\star)\cdot N^{3/4}\cdot\text{polylog}(N+n(\underline{d}^\star)+d_0)+d_0^2\cdot\log N+d_0 n(\underline{d}^\star)\log(n(\underline{d}^\star)+N+d_0)^{3/2}\right)}{N\cdot\mathbb{P}_S(\text{consistent } S)} \\ &= \frac{1}{\mathbb{P}_S\left(\text{consistent } S\right)}\cdot\left(H\left(\varepsilon^\star\right) + O\left(C_{\min}\left(N, d_0, \underline{d}^\star\right)\right)\right) .\end{aligned}$$

Rearranging the above inequality and recalling Remark 20, we have,

$$\mathbb{E}_S\left[\mathcal{L}_{\mathcal{D}}\left(A\left(S\right)\right) \mid \text{consistent } S\right] \leq 1 - 2^{-\frac{H(\varepsilon^\star)}{\mathbb{P}_S(\text{consistent } S)} - O(C_{\min}(N, d_0, \underline{d}^\star))} .$$

Then, using Lemma 15, we get,

$$\mathbb{E}_S\left[\mathcal{L}_{\mathcal{D}}\left(A\left(S\right)\right) \mid \text{consistent } S\right] \leq 1 - 2^{-\frac{H(\varepsilon^\star)}{\mathbb{P}_S(\text{consistent } S)}} + O\left(C_{\min}\left(N, d_0, \underline{d}^\star\right)\right) .$$

The bound is derived using the following observation. Since for a RV X in $[0,1]$ and a binary RV Y we have

$$\mathbb{E}[X] = \mathbb{E}[X \mid Y]\underbrace{\mathbb{P}(Y)}_{\leq 1} + \underbrace{\mathbb{E}[X \mid \neg Y]}_{\leq 1}\mathbb{P}(\neg Y) \leq \mathbb{E}[X \mid Y] + \mathbb{P}[\neg Y] ,$$

we conclude the proof as

$$\mathbb{E}_S\left[\mathcal{L}_{\mathcal{D}}\left(A\left(S\right)\right)\right] \leq \mathbb{E}_S\left[\mathcal{L}_{\mathcal{D}}\left(A\left(S\right)\right) \mid \text{consistent } S\right] + \mathbb{P}\left(\text{inconsistent } S\right) .$$

For the independent noise setting, we combine Lemma 22 and Lemma 57 to get

$$\begin{aligned}
&\left|\mathbb{E}_S\left[\mathcal{L}_{\mathcal{D}}\left(A\left(S\right)\right) \mid \text{consistent } S\right] - 2\varepsilon^\star\left(1-\varepsilon^\star\right)\right| \\
&\qquad\qquad \leq \left(1-2\varepsilon^\star\right) O\left(\sqrt{C\left(N\right)}\right) + \frac{\left(N-1\right)\mathcal{D}_{\max}}{3} ,
\end{aligned}$$

where

$$\begin{aligned}
C\left(N\right) &= \frac{I\left(S; A\left(S\right)\right) - N \cdot \left(H\left(\varepsilon^\star\right) - \mathbb{P}\left(\text{inconsistent } S\right)\right)}{N\left(1 - \mathbb{P}\left(\text{inconsistent } S\right)\right)} \\
&\leq \frac{w(\underline{d}^\star) + N \cdot H(\varepsilon^\star) + O(\delta(N, d_0, \underline{d}^\star)) - N \cdot (H(\varepsilon^\star) - \mathbb{P}(\text{inconsistent } S))}{N(1 - \mathbb{P}(\text{inconsistent } S))} \\
&= \frac{O\left(\frac{w(\underline{d}^\star) + \delta(N, d_0, \underline{d}^\star)}{N}\right) + \mathbb{P}\left(\text{inconsistent } S\right)}{\mathbb{P}\left(\text{consistent } S\right)} \\
&= \frac{O\left(C_{\min}\left(N, d_0, \underline{d}^\star\right)\right) + \mathbb{P}\left(\text{inconsistent } S\right)}{\mathbb{P}\left(\text{consistent } S\right)}
\end{aligned}$$

Finally, using the inequality from Lemma 23, we have,

$$\begin{aligned}
&\left|\mathbb{E}_{S,A(S)}\left[\mathcal{L}_{\mathcal{D}}\left(A\left(S\right)\right)\right] - 2\varepsilon^\star\left(1-\varepsilon^\star\right)\right| \\
&\qquad \leq \left|\mathbb{E}_{S,A(S)}\left[\mathcal{L}_{\mathcal{D}}\left(A\left(S\right)\right) \mid \text{consistent } S\right] - 2\varepsilon^\star\left(1-\varepsilon^\star\right)\right| + \mathbb{P}(\text{inconsistent } S)
\end{aligned}$$

■

G.2. Derivation of the posterior sampling generalization bounds (Section 4.2)

Lemma 59 *For the posterior sampling algorithm*

$$I\left(S;A\left(S\right)\right)\leq\mathbb{E}_{S}\left[\log\left(\frac{1}{p_{S}}\right)\middle|\text{consistent }S\right]\mathbb{P}_{S}\left(\text{consistent }S\right)+\frac{2}{e\ln 2}\,.$$

Proof Recall the definition of the marginal distribution of the algorithm's output (a hypothesis h) is

$$d\nu\left(h\right)=\sum_{s}dp\left(s,h\right)\ ,$$

where s are all possible realizations of a (training) sample of size N.

For $h=\star$, we have $d\nu\left(\star\right)=\mathbb{P}_{S}\left(\text{inconsistent }S\right)$.

For $h\neq\star$, since $\mathcal{L}_{s}\left(h\right)=0$ implies that s is consistent, we have

$$\begin{aligned}
d\nu\left(h\right)&\triangleq\sum_{s}dp\left(s,h\right)=\sum_{s}\frac{\mathbb{I}\left\{\mathcal{L}_{s}\left(h\right)=0\right\}}{p_{s}}d\mathcal{P}\left(h\right)d\mathcal{D}^{N}\left(s\right)\\
&=\sum_{s:p_{s}>0}\frac{\mathbb{I}\left\{\mathcal{L}_{s}\left(h\right)=0\right\}}{p_{s}}d\mathcal{P}\left(h\right)d\mathcal{D}^{N}\left(s\right)\\
&=d\mathcal{P}\left(h\right)\sum_{s:p_{s}>0}\frac{\mathbb{I}\left\{\mathcal{L}_{s}\left(h\right)=0\right\}}{p_{s}}d\mathcal{D}^{N}\left(s\right)\\
&=d\mathcal{P}\left(h\right)\mathbb{E}_{S\sim\mathcal{D}^{N}}\left[\frac{\mathbb{I}\left\{p_{S}>0\right\}}{p_{S}}\mathbb{I}\left\{\mathcal{L}_{S}\left(h\right)=0\right\}\right]\,.
\end{aligned}$$

where, for ease of notation, we use the convention that $\frac{\mathbb{I}\{p_s>0\}}{p_s}=0$ when $p_s=0$. Denoting

$$\pi\left(h\right)\triangleq\mathbb{E}_{S\sim\mathcal{D}^{N}}\left[\frac{\mathbb{I}\left\{p_{S}>0\right\}}{p_{S}}\mathbb{I}\left\{\mathcal{L}_{S}\left(h\right)=0\right\}\right]\ ,$$

we get

$$d\nu\left(h\right)=d\mathcal{P}\left(h\right)\pi\left(h\right)\ .$$

Notice that if there exists some $s\in\operatorname{supp}\left(\mathcal{D}^{N}\right)$ such that $\mathcal{L}_{s}\left(h\right)=0$ then $\pi\left(h\right)>0$. Using the definition of the mutual information:

$$\begin{aligned}
&I\left(S;A\left(S\right)\right)=\sum_{s}\sum_{h\in\mathcal{H}\cup\{\star\}}dp\left(s,h\right)\log\left(\frac{dp\left(s,h\right)}{d\nu\left(h\right)d\mathcal{D}\left(s\right)}\right)\\
&=\sum_{s:p_{s}=0}dp\left(s,\star\right)\log\left(\frac{dp\left(s,\star\right)}{d\nu\left(\star\right)d\mathcal{D}\left(s\right)}\right)+\sum_{s:p_{s}>0}\sum_{h\in\mathcal{H}}dp\left(s,h\right)\log\left(\frac{dp\left(s,h\right)}{d\nu\left(h\right)d\mathcal{D}\left(s\right)}\right)\\
&=\sum_{s:p_{s}=0}d\mathcal{D}\left(s\right)\log\left(\frac{d\mathcal{D}\left(s\right)}{\mathbb{P}_{S}\left(\text{inconsistent }S\right)d\mathcal{D}\left(s\right)}\right)+\\
&\quad\sum_{s:p_{s}>0}\sum_{h:\mathcal{L}_{s}(h)=0}\frac{1}{p_{s}}d\mathcal{P}\left(h\right)d\mathcal{D}\left(s\right)\log\left(\frac{\frac{1}{p_{s}}d\mathcal{P}\left(h\right)d\mathcal{D}\left(s\right)}{d\mathcal{P}\left(h\right)\pi\left(h\right)d\mathcal{D}\left(s\right)}\right)\\
&=\sum_{s:p_{s}=0}d\mathcal{D}\left(s\right)\log\left(\frac{1}{\mathbb{P}_{S}(\text{inconsistent }S)}\right)+\sum_{s:p_{s}>0}\sum_{h:\mathcal{L}_{s}(h)=0}\frac{1}{p_{s}}d\mathcal{P}\left(h\right)d\mathcal{D}\left(s\right)\log\left(\frac{1}{p_{s}\pi\left(h\right)}\right)\,.
\end{aligned}$$

Simplifying each term separately, the first sum immediately simplifies to

$$-\mathbb{P}_S\left(\text{inconsistent } S\right)\log\left(\mathbb{P}_S\left(\text{inconsistent } S\right)\right) \leq \frac{1}{e\ln 2}\,,$$

and

$$\begin{aligned}
&\sum_{s:p_s>0}\sum_{h:\mathcal{L}_s(h)=0}\frac{1}{p_s}d\mathcal{P}\left(h\right)d\mathcal{D}\left(s\right)\log\left(\frac{1}{p_s\pi\left(h\right)}\right)\\
&=-\sum_{s:p_s>0}\sum_{h:\mathcal{L}_s(h)=0}\frac{1}{p_s}d\mathcal{P}\left(h\right)d\mathcal{D}\left(s\right)\log\left(p_s\right)-\sum_{s:p_s>0}\sum_{h:\mathcal{L}_s(h)=0}\frac{1}{p_s}d\mathcal{P}\left(h\right)d\mathcal{D}\left(s\right)\log\left(\pi\left(h\right)\right)\\
&=-\sum_{s:p_s>0}\frac{1}{p_s}\log\left(p_s\right)d\mathcal{D}\left(s\right)\underbrace{\sum_{h:\mathcal{L}_s(h)=0}d\mathcal{P}\left(h\right)}_{=p_s}\\
&\quad-\sum_{s:p_s>0}\sum_{h:\pi(h)>0}\frac{\mathbb{I}\left\{\mathcal{L}_s\left(h\right)=0\right\}}{p_s}d\mathcal{P}\left(h\right)d\mathcal{D}\left(s\right)\log\left(\pi\left(h\right)\right)\\
&=-\sum_{s:p_s>0}\frac{1}{p_s}\log\left(p_s\right)d\mathcal{D}\left(s\right)p_s-\sum_{h:\pi(h)>0}\log\left(\pi\left(h\right)\right)d\mathcal{P}\left(h\right)\underbrace{\sum_{s:p_s>0}\frac{\mathbb{I}\left\{\mathcal{L}_s\left(h\right)=0\right\}}{p_s}d\mathcal{D}\left(s\right)}_{=\pi(h)}\\
&=-\sum_{s:p_s>0}\log\left(p_s\right)d\mathcal{D}\left(s\right)-\sum_{h:\pi(h)>0}\pi\left(h\right)\log\left(\pi\left(h\right)\right)d\mathcal{P}\left(h\right)\\
&=-\mathbb{E}_S\left[\log\left(p_S\right)\mathbb{I}\left\{p_S>0\right\}\right]-\mathbb{E}_{h\sim\mathcal{P}}\left[\mathbb{I}\left\{\pi\left(h\right)>0\right\}\pi\left(h\right)\log\left(\pi\left(h\right)\right)\right]\\
&=\mathbb{E}_S\left[\log\left(\frac{1}{p_S}\right)\mid p_S>0\right]\mathbb{P}_S\left(p_S>0\right)+\underbrace{\mathbb{E}_{h\sim\mathcal{P}}\left[-\pi\left(h\right)\log\left(\pi\left(h\right)\right)\mathbb{I}\left\{\pi\left(h\right)>0\right\}\right]}_{\leq 1/e\ln 2}\\
&\leq\mathbb{E}_S\left[\log\left(\frac{1}{p_S}\right)\mid\text{consistent } S\right]\mathbb{P}_S\left(\text{consistent } S\right)+\frac{1}{e\ln 2}\,.
\end{aligned}$$

Putting all of this together,

$$I\left(S;A\left(S\right)\right)\leq\mathbb{E}_S\left[\log\left(\frac{1}{p_S}\right)\middle|\text{consistent } S\right]\mathbb{P}_S\left(\text{consistent } S\right)+\frac{2}{e\ln 2}\,.$$

■

Corollary 60 *The generalization of posterior sampling satisfies*

$$-\log\left(1-\mathbb{E}_{S,A(S)}\left[\mathcal{L}_{\mathcal{D}}\left(A\left(S\right)\right)\mid\text{consistent }S\right]\right)\leq\frac{\mathbb{E}_{S}\left[\log\left(\frac{1}{p_{S}}\right)\middle|\text{consistent }S\right]+3}{N}.$$

Proof Combining Lemma 21 and Lemma 59 we get

$$I\left(S;A\left(S\right)\right)\geq-N\log\left(1-\mathbb{E}_{S,A(S)}\left[\mathcal{L}_{\mathcal{D}}\left(A\left(S\right)\right)\mid\text{consistent }S\right]\right)\mathbb{P}_{S}\left(\text{consistent }S\right)$$

and

$$I\left(S;A\left(S\right)\right)\leq\mathbb{E}_{S}\left[\log\left(\frac{1}{p_{S}}\right)\middle|\text{consistent }S\right]\mathbb{P}_{S}\left(\text{consistent }S\right)+\frac{2}{e\ln 2}$$

so

$$\begin{aligned}&-N\log\left(1-\mathbb{E}_{S,A(S)}\left[\mathcal{L}_{\mathcal{D}}\left(A\left(S\right)\right)\mid\text{consistent }S\right]\right)\mathbb{P}_{S}\left(\text{consistent }S\right)\\&\qquad\leq\mathbb{E}_{S}\left[\log\left(\frac{1}{p_{S}}\right)\middle|\text{consistent }S\right]\mathbb{P}_{S}\left(\text{consistent }S\right)+\frac{2}{e\ln 2}\end{aligned}$$

and finally, using $2/e\ln 2\leq 1.5$ and recalling 20 we get

$$-\log\left(1-\mathbb{E}_{S,A(S)}\left[\mathcal{L}_{\mathcal{D}}\left(A\left(S\right)\right)\mid\text{consistent }S\right]\right)\leq\frac{\mathbb{E}_{S}\left[\log\left(\frac{1}{p_{S}}\right)\middle|\text{consistent }S\right]+3}{N}.$$

■

Let $\bar{h}$ be a network with depth L, dimensions $\underline{\bar{d}}$, and parameters $\bar{\boldsymbol{\theta}} = \left\{\bar{\mathbf{W}}_l, \bar{\mathbf{b}}_l, \bar{\gamma}_l\right\} \in \Theta^{\text{BTN}}\left(\underline{\bar{d}}\right)$. Let $\underline{d} \geq \underline{\bar{d}}$. Similar to $\Theta^{\text{BTN}}\left(\underline{d}; h_1, h_2\right)$ introduced in Lemma 51, let $\Theta^{\text{BTN}}\left(\underline{d}; \bar{h}\right) \subset \Theta^{\text{BTN}}\left(\underline{d}\right)$ be the set of parameters $\boldsymbol{\theta}$ that implement $\bar{h}$ by setting a subset of the parameters to be equal to $\bar{\boldsymbol{\theta}}$, and zero the effect of redundant neurons by setting their bias and neuron scaling terms to be 0. This is illustrated in Figure 4. In particular, in our notation, $\Theta^{\text{BTN}}\left(\underline{d}; h_1, h_2\right) = \Theta^{\text{BTN}}\left(\underline{d}; h_1 \oplus h_2\right)$.

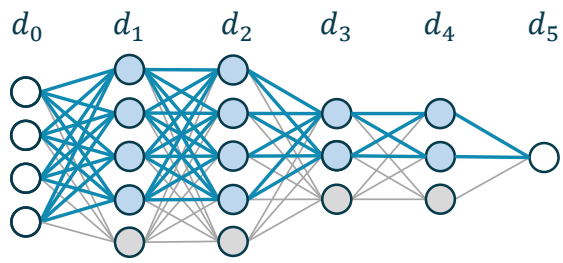


Figure 4: **Implementing a narrow network with a wider network.** Blue edges represent parameters set to equal the parameters of $\bar{h}$, grey edges represent unconstrained parameters.

Lemma 61 *Let h be a network with depth L and dimensions $\underline{\bar{d}}$. Let $\underline{d} \geq \underline{\bar{d}}$. Then*

$$-\log\left(\frac{\left|\Theta^{BTN}\left(\underline{d}; \bar{h}\right)\right|}{\left|\Theta^{BTN}\left(\underline{d}\right)\right|}\right) \leq w\left(\underline{\bar{d}}\right) + O\left(n\left(\underline{d}\right) \cdot \log\left(\underline{d}_{\max} + d_0\right)\right) .$$

Proof We prove this by counting the number of constrained parameters in $\Theta^{\text{BTN}}\left(\underline{d}; \bar{h}\right)$. The number of constrained weights is

$$\bar{d}_1 d_0 + \sum_{l=2}^{L} \bar{d}_l \bar{d}_{l-1} ,$$

which is exactly $w\left(\underline{\bar{d}}\right)$. In addition, there are $n\left(\underline{d}\right)$ constrained bias terms, and $n\left(\underline{d}\right)$ constrained scaling terms. In total, after accounting for the quantization of each parameter, this means that

$$\frac{\left|\Theta^{\text{BTN}}\left(\underline{d}; \bar{h}\right)\right|}{\left|\Theta^{\text{BTN}}\left(\underline{d}\right)\right|} \geq \left(\underbrace{2^{w\left(\underline{\bar{d}}\right)}}_{\text{weights}} \cdot \underbrace{3^{n\left(\underline{d}\right)}}_{\text{scaling terms}} \cdot \prod_{l=1}^{L} \left(2d_{l-1}\right)^{d_l}\right)^{-1}$$

so

$$\begin{aligned}
&-\log\left(\frac{\left|\Theta^{\text{BTN}}\left(\underline{d}; \bar{h}\right)\right|}{\left|\Theta^{\text{BTN}}\left(\underline{d}\right)\right|}\right) \\
&\leq w\left(\underline{\bar{d}}\right) + n\left(\underline{d}\right) \cdot \log 3 + \sum_{l=1}^{L} \bar{d}_l \cdot \log\left(2d_{l-1}\right) \\
&\leq w\left(\underline{\bar{d}}\right) + n\left(\underline{d}\right) \cdot \log 3 + n\left(\underline{d}\right) \cdot \log\left(2\underline{d}_{\max} + 2d_0\right) \\
&= w\left(\underline{\bar{d}}\right) + O\left(n\left(\underline{d}\right) \cdot \log\left(\underline{d}_{\max} + d_0\right)\right) .
\end{aligned}$$

■

Combining Lemma 61 with Assumption 2, and Corollary 4 gives the following lemma.

Lemma 62 *Consider a distribution $\mathcal{D}$ induced by a noisy teacher model of depth $L^\star$ and widths $\underline{d}^\star$ (Assumption 2) with a noise level of $\varepsilon^\star < 1/2$. Let $S \sim \mathcal{D}^N$ be a training set with effective training set label noise $\hat{\varepsilon}_{\text{tr}}$ as defined in (2). Then there exist constants $c_1, c_2 > 0$ such that for any student network of depth $L \geq \max\{L^\star, 14\} + 2$ and widths $\underline{d} \in \mathbb{N}^L$ satisfying*

$$\forall l = 1, \ldots, L^\star - 1 \quad d_l \geq d_l^\star + N^{3/4} \cdot (\log N)^{c_1} + c_2 \cdot d_0 \cdot \log(N) \ ,$$

it holds for posterior sampling with a uniform prior over hypotheses that

$$\begin{aligned}
&\mathbb{E}_S \left[\log\left(\frac{1}{p_S}\right) \mid \textit{consistent } S \right] \\
&\quad \leq w(\underline{d}^\star) + N \cdot H(\hat{\varepsilon}_{\text{tr}}) + 2n(\underline{d}^\star) N^{3/4} \text{polylog} N \\
&\qquad + O\left(d_0 (d_0 + n(\underline{d}^\star)) \cdot \log(N) + n(\underline{d}) \cdot \log(\underline{d}_{\max} + d_0)\right) \ .
\end{aligned}$$

Proof Notice that for posterior sampling with uniform distribution over parameters, the interpolation probability p_S can be lower bounded as

$$p_S \geq \frac{\left|\Theta^{\text{BTN}}\left(\underline{d}; h^\star \oplus \tilde{h}_S\right)\right|}{\left|\Theta^{\text{BTN}}(\underline{d})\right|}$$

and therefore

$$\log\left(\frac{1}{p_S}\right) \leq -\log\left(\frac{\left|\Theta^{\text{BTN}}\left(\underline{d}; h^\star \oplus \tilde{h}_S\right)\right|}{\left|\Theta^{\text{BTN}}(\underline{d})\right|}\right) \ .$$

Then, using the bounds from Lemma 61 with the one from Corollary 4

$$\begin{aligned}
\log\left(\frac{1}{p_S}\right) &\leq w(\underline{d}^\star) + N \cdot H(\mathcal{L}_S(h^\star)) + 2n(\underline{d}^\star) N^{3/4} \text{polylog} N + O\left(d_0 (d_0 + n(\underline{d}^\star)) \cdot \log N\right) \\
&\quad + O\left(n(\underline{d}) \cdot \log(\underline{d}_{\max} + d_0)\right) \\
&= w(\underline{d}^\star) + N \cdot H(\mathcal{L}_S(h^\star)) + 2n(\underline{d}^\star) N^{3/4} \text{polylog} N \\
&\quad + O\left(d_0 (d_0 + n(\underline{d}^\star)) \cdot \log(N) + n(\underline{d}) \cdot \log(\underline{d}_{\max} + d_0)\right) \ .
\end{aligned}$$

By taking the expectation and using Jensen's inequality with the concave H we arrive at

$$\mathbb{E}_S\left[H(\mathcal{L}_S(h^\star)) \mid \text{consistent } S\right] \leq H\left(\mathbb{E}_S\left[\mathcal{L}_S(h^\star) \mid \text{consistent } S\right]\right) = H(\hat{\varepsilon}_{\text{tr}})$$

and hence

$$\begin{aligned}
&\mathbb{E}_S \left[\log\left(\frac{1}{p_S}\right) \middle| \text{consistent } S \right] \\
&\leq w(\underline{d}^\star) + N \cdot H(\hat{\varepsilon}_{\text{tr}}) + 2n(\underline{d}^\star) N^{3/4} \text{polylog} N \\
&\quad + O\left(d_0 (d_0 + n(\underline{d}^\star)) \cdot \log(N) + n(\underline{d}) \cdot \log(\underline{d}_{\max} + d_0)\right) \ .
\end{aligned}$$

■

Recall Theorem 7. Consider a distribution $\mathcal{D}$ induced by a noisy teacher model of depth $L^\star$ and widths $\underline{d}^\star$ (Assumption 2) with a noise level of $\varepsilon^\star < 1/2$. Let $S \sim \mathcal{D}^N$ be a training set such that $N = o(\sqrt{1/\mathcal{D}_{\max}})$. Then, there exist constants $c_1, c_2 > 0$ such that for any student network of depth $L \geq \max\{L^\star, 14\} + 2$ and widths $\underline{d} \in \mathbb{N}^L$ holding

$$\forall l = 1, \ldots, L^\star - 1 \quad d_l \geq d_l^\star + N^{3/4} \cdot (\log N)^{c_1} + c_2 \cdot d_0 \cdot \log(N) \ , \tag{9}$$

the generalization error of posterior sampling satisfies the following.

- **Under arbitrary label noise**,

$$\mathbb{E}_{S,A(S)}\left[\mathcal{L}_{\mathcal{D}}(A(S))\right] \leq 1 - 2^{-H(\varepsilon^\star)} + 2N^2\mathcal{D}_{\max} + O\left(C_{\text{rand}}(N)\right) \ .$$

- **Under independent label noise,**

$$\begin{aligned}&\left|\mathbb{E}_{S,A(S)}\left[\mathcal{L}_{\mathcal{D}}(A(S))\right] - 2\varepsilon^\star(1-\varepsilon^\star)\right| \\ &\leq (1-2\varepsilon^\star)\sqrt{\frac{O\left(C_{\text{rand}}(N)\right) + \mathbb{P}\left(\text{inconsistent } S\right)}{\mathbb{P}\left(\text{consistent } S\right)}} + \frac{(N-1)\,\mathcal{D}_{\max}}{3} + \mathbb{P}\left(\text{inconsistent } S\right) \ ,\end{aligned}$$

where

$$C_{\text{rand}}(N) = \frac{n(\underline{d}^\star)\cdot\text{polylog}(N)}{\sqrt[4]{N}} + \frac{w(\underline{d}^\star) + d_0\left(d_0 + n(\underline{d}^\star)\right)\cdot\log(N) + n(\underline{d})\cdot\log\left(\underline{d}_{\max} + d_0\right)}{N} \ .$$

Remark 63 *The bound shown in Section 4.2 is found by bounding* $\mathbb{P}\left(\text{inconsistent } S\right)$ *as in Lemma 18. Assuming that* $N = \omega\left(n(\underline{d}^\star)^4\,\text{polylog}\left(n(\underline{d}^\star)\right) + d_0^2\log d_0\right)$ *we can deduce that* $N = \omega\left(w(\underline{d}^\star)\right)$ *as well since*

$$w(\underline{d}^\star) \leq \left(n(\underline{d}^\star) + d_0\right)^2 \leq 4\left(\max\left\{n(\underline{d}^\star), d_0\right\}\right)^2 \ .$$

Together with $N = o\left(\sqrt{1/\mathcal{D}_{\max}}\right)$ *we get the desired form of the bounds.*

Proof Corollary 4 implies that there exist $c_1, c_2 > 0$ such that a student NN satisfying (9) can interpolate any consistent dataset, and so posterior sampling is interpolating for all consistent datasets.

We start by proving the bound for arbitrary label noise.

First, we notice that

$$\begin{aligned}\hat{\varepsilon}_{\text{tr}} = \mathbb{P}(Y_1 \neq h^\star(X_1) \mid \text{consistent } S) &= \frac{\mathbb{P}(Y_1 \neq h^\star(X_1), \text{consistent } S)}{\mathbb{P}\left(\text{consistent } S\right)} \\ &\leq \frac{\mathbb{P}(Y_1 \neq h^\star(X_1))}{\mathbb{P}\left(\text{consistent } S\right)} = \frac{\varepsilon^\star}{\mathbb{P}\left(\text{consistent } S\right)} \ .\end{aligned}$$

The entropy function H is increasing in $\left[0, \frac{1}{2}\right]$ and achieves its maximum at $\frac{1}{2}$, so together with the inequality above, we get,

$$\begin{aligned} H\left(\hat{\varepsilon}_{\text{tr}}\right) &\leq H\left(\min\left\{\frac{\varepsilon^\star}{\mathbb{P}(\text{consistent } S)}, \frac{1}{2}\right\}\right) = H\left(\varepsilon^\star + \min\left\{\frac{\varepsilon^\star}{\mathbb{P}(\text{consistent } S)} - \varepsilon^\star, \frac{1}{2} - \varepsilon^\star\right\}\right) \\ &= H\Big(\varepsilon^\star + \underbrace{\min\left\{\varepsilon^\star \frac{\mathbb{P}(\text{inconsistent } S)}{\mathbb{P}(\text{consistent } S)}, \frac{1}{2} - \varepsilon^\star\right\}}_{\triangleq \Delta}\Big) = H\left(\varepsilon^\star + \Delta\right) . \end{aligned}$$

Employing the concavity of the entropy function, we get,

$$H\left(\hat{\varepsilon}_{\text{tr}}\right) \leq H\left(\varepsilon^\star + \Delta\right) \leq H(\varepsilon^\star) + H'(\varepsilon^\star) \cdot \Delta \leq H(\varepsilon^\star) + \underbrace{H'(\varepsilon^\star) \cdot \varepsilon^\star}_{\leq \frac{1}{2}, \text{algebraically}} \cdot \frac{\mathbb{P}\left(\text{inconsistent } S\right)}{\mathbb{P}\left(\text{consistent } S\right)} .$$

Then, combining the above with Corollary 60, Lemma 62, we have that

$$\begin{aligned} &-\log\left(1 - \mathbb{E}_{(S,A(S))}\left[\mathcal{L}_{\mathcal{D}}\left(A\left(S\right)\right) \mid \text{consistent } S\right]\right) \leq \frac{\mathbb{E}_S\left[\log\left(1/p_S\right) \mid \text{consistent } S\right] + 3}{N} \\ &\quad \leq H\left(\hat{\varepsilon}_{\text{tr}}\right) + \frac{1}{N}\Big(w\left(\underline{d}^\star\right) + N \cdot H\left(\hat{\varepsilon}_{\text{tr}}\right) + 2n\left(\underline{d}^\star\right) N^{3/4} \text{polylog} N \\ &\qquad + O\left(d_0\left(d_0 + n\left(\underline{d}^\star\right)\right) \cdot \log\left(N\right) + n\left(\underline{d}\right) \cdot \log\left(\underline{d}_{\max} + d_0\right)\right)\Big) \\ &\quad \leq H\left(\hat{\varepsilon}_{\text{tr}}\right) + O\left(\frac{n(\underline{d}^\star) \cdot \text{polylog}(N)}{\sqrt[4]{N}} + \frac{w(\underline{d}^\star) + d_0(d_0 + n(\underline{d}^\star)) \cdot \log(N) + n(\underline{d}) \cdot \log(\underline{d}_{\max} + d_0)}{N}\right) \\ &\quad \leq H(\varepsilon^\star) + \frac{\mathbb{P}(\text{inconsistent } S)}{2\mathbb{P}(\text{consistent } S)} \\ &\qquad + O\left(\frac{n(\underline{d}^\star) \cdot \text{polylog}(N)}{\sqrt[4]{N}} + \frac{w(\underline{d}^\star) + d_0(d_0 + n(\underline{d}^\star)) \cdot \log(N) + n(\underline{d}) \cdot \log(\underline{d}_{\max} + d_0)}{N}\right) \\ &\quad = H(\varepsilon^\star) + \frac{\mathbb{P}(\text{inconsistent } S)}{2\mathbb{P}(\text{consistent } S)} + O\left(C_{\text{rand}}\left(N\right)\right) . \end{aligned}$$

Rearranging the inequality results in

$$\begin{aligned} &\mathbb{E}_{(S,A(S))}\left[\mathcal{L}_{\mathcal{D}}\left(A\left(S\right)\right) \mid \text{consistent } S\right] \\ &\qquad \leq 1 - 2^{-H(\varepsilon^\star) - \frac{\mathbb{P}(\text{inconsistent } S)}{2\mathbb{P}(\text{consistent } S)} - O(C_{\text{rand}}(N))} \end{aligned}$$

Then, using Lemma 15, we get,

$$\begin{aligned} &\mathbb{E}_{(S,A(S))}\left[\mathcal{L}_{\mathcal{D}}\left(A\left(S\right)\right) \mid \text{consistent } S\right] \\ &\leq 1 - 2^{-H(\varepsilon^\star)} + \frac{\mathbb{P}\left(\text{inconsistent } S\right)}{2\mathbb{P}\left(\text{consistent } S\right)} + O\left(C_{\text{rand}}\left(N\right)\right) . \end{aligned}$$

Repeating the argument from the proof of Theorem 6, since for an RV X in $[0, 1]$ and a binary RV Y we have

$$\mathbb{E}[X] = \mathbb{E}[X \mid Y] \underbrace{\mathbb{P}(Y)}_{\leq 1} + \underbrace{\mathbb{E}[X \mid \neg Y]}_{\leq 1} \mathbb{P}(\neg Y) \leq \mathbb{E}[X \mid Y] + \mathbb{P}(\neg Y) ,$$

we have,

$$\begin{aligned}
&\mathbb{E}_{(S,A(S))}\left[\mathcal{L}_{\mathcal{D}}\left(A\left(S\right)\right)\right] \leq \mathbb{E}_{(S,A(S))}\left[\mathcal{L}_{\mathcal{D}}\left(A\left(S\right)\right) \mid \text{consistent } S\right] + \mathbb{P}\left(\text{inconsistent } S\right) \\
&\leq 1 - 2^{-H(\varepsilon^\star)} + \frac{\mathbb{P}\left(\text{inconsistent } S\right)}{2\mathbb{P}\left(\text{consistent } S\right)} + \mathbb{P}\left(\text{inconsistent } S\right) + O\left(C_{\text{rand}}\left(N\right)\right) \\
&\leq 1 - 2^{-H(\varepsilon^\star)} + 2\frac{\mathbb{P}\left(\text{inconsistent } S\right)}{\mathbb{P}\left(\text{consistent } S\right)} + O\left(C_{\text{rand}}\left(N\right)\right) \\
&\leq 1 - 2^{-H(\varepsilon^\star)} + 2\frac{\frac{1}{2}N^2\mathcal{D}_{\max}}{1 - \frac{1}{2}N^2\mathcal{D}_{\max}} + O\left(C_{\text{rand}}\left(N\right)\right) \\
&\leq 1 - 2^{-H(\varepsilon^\star)} + 2N^2\mathcal{D}_{\max} + O\left(C_{\text{rand}}\left(N\right)\right)
\end{aligned}$$

where in the last inequality we used $t/\left(1-t\right) \leq 2t$ for $t \in [0, 1/2]$.

Moving on to the independent noise setting, we combine Lemma 59, Lemma 62, and $\hat{\varepsilon}_{\text{tr}} \leq \varepsilon^\star < \frac{1}{2}$ from Lemma 19, to bound the mutual information as

$$\begin{aligned}
I\left(S; A\left(S\right)\right) \leq\ & \mathbb{E}_S\left[\log\left(\frac{1}{p_S}\right) \mid \text{consistent } S\right] \overbrace{\mathbb{P}_S\left(\text{consistent } S\right)}^{\leq 1} + \frac{2}{e\ln 2} \\
\leq\ & \mathbb{E}_S\left[\log\left(\frac{1}{p_S}\right) \mid \text{consistent } S\right] + 1.1 \\
\leq\ & N \cdot H\left(\varepsilon^\star\right) + O\left(N \cdot C_{\text{rand}}\left(N\right)\right) .
\end{aligned}$$

Plugging the above into $C(N)$ of Lemma 22, we get,

$$\begin{aligned}
C\left(N\right) &= \frac{I\left(S; A\left(S\right)\right) - N \cdot H\left(\varepsilon^\star\right) + N \cdot \mathbb{P}_{S\sim\mathcal{D}^N}\left(\text{inconsistent } S\right)}{N \cdot \mathbb{P}\left(\text{consistent } S\right)} \\
&\leq \frac{N \cdot H(\varepsilon^\star) + O(N \cdot C_{\text{rand}}(N)) - N \cdot H(\varepsilon^\star) + N \cdot \mathbb{P}(\text{inconsistent } S)}{N \cdot \mathbb{P}(\text{consistent } S)} \\
&= \frac{O\left(C_{\text{rand}}\left(N\right)\right) + \mathbb{P}\left(\text{inconsistent } S\right)}{\mathbb{P}\left(\text{consistent } S\right)} .
\end{aligned}$$

Then we continue as in the arbitrary noise setting to get the desired bound. ■